



**António José
Claro Ferreira**

**Sistema de Identificação pessoal baseado em sinais
de EEG**



**António José
Claro Ferreira**

Sistema de Identificação pessoal baseado em sinais de EEG

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica da Dra. Pétia Georgieva, e da Dra. Ana Maria Tomé, professoras do Departamento de Electrónica Telecomunicações e Informática da Universidade de Aveiro

Dedico este trabalho aos meus familiares e amigos que me ajudaram na longa etapa acadêmica.

o júri

presidente

Prof. Doutor José Fernando Rocha Pereira
Universidade de Aveiro

vogais

Arguente Principal: Doutor Luís Miguel Almeida da Silva
Departamento de Matemática para Ciência e Tecnologia da Escola de Ciências da Universidade do Minho

Orientadora: Professora Doutora Pétia Georgieva Georgieva
Universidade de Aveiro

Orientadora: Professora Doutora Ana Maria Perfeito Tomé
Universidade de Aveiro

agradecimentos

Gostava de agradecer a todas as pessoas que me apoiaram não só durante todos estes meses de trabalho árduo, para fazer esta tese, mas também aqueles que me ajudaram ao longo do curso.

Gostaria deixar um agradecimento especial às orientadoras Prof. Dra. Pétia Georgieva e Prof. Dra. Ana Maria Tomé, por todo o seu apoio, paciência, simpatia e disponibilidade. O meu obrigado por me terem ajudado nesta derradeira etapa académica.

Fica ainda uma palavra de agradecimento aos meus companheiros de curso e verdadeiros amigos que tive o prazer de fazer durante estes últimos e melhores anos da minha vida. Agradeço também aos meus amigos de longa data de Braga, que têm sido uma constante de motivação e exemplo ao longo dos anos.

Por fim agradeço às pessoas mais importantes, que são a minha família, pois é graças a eles que hoje sou aquilo que sou.

palavras-chave

Identificação e Autenticação pessoal, Electroencefalograma, Cérebro, Potencias Evocados Visuais.

resumo

O objectivo principal desta tese é desenvolver um sistema de identificação pessoal baseado em sinais EEG. Os sinais de EEG são adquiridos através de 22 eléctrodos colocados no escalpe (20 para EEG e 2 para EOG), enquanto o individuo for sujeito a estímulos visuais. A viabilidade deste sistema biométrico é testada com os dados de 16 indivíduos implementando vários métodos de classificação e processamento de sinal.

No primeiro capítulo é feita uma introdução ao conceito de Identificação Biométrica e o potencial deste modelo para um futuro próximo. Neste capítulo são também indicados os objectivos da tese.

No segundo capítulo são apresentados alguns dos conceitos teóricos, necessários para uma melhor compreensão do trabalho em questão. São também descritos alguns trabalhos relacionados que serviram de ponto de partida para o trabalho realizado.

No terceiro capítulo, Realização Experimental, é exposto a metodologia seguida para se atingirem os resultados. São descritos todos os processos que levaram à obtenção dos valores apresentados no capítulo seguinte, desde a identificação das características individuais, até às classificações.

No quarto capítulo expuseram-se os resultados obtidos para os diferentes classificadores abordados, com e sem PCA e nas suas vertentes lineares e não lineares. Os classificadores usados foram o MVSVM (Majority Voting Support Vector Machine) e o OAASVM (One Against All Support Vector Machine).

No quinto e último capítulo, apresentaram-se as conclusões onde, com base nos resultados, se concluiu que: a rejeição das épocas com artefactos melhora os resultados finais; que a PCA aplicada aos sinais EEG disponíveis contribui para o aumento do erro da classificação; os classificadores na sua vertente não linear são mais eficazes que os lineares; que o treino e teste dos classificadores devem ser feitos com dados provenientes de todas as experiências realizadas para obtenção dos PEVs; o número de épocas ideal para o teste de um classificador não linear fica algures entre o intervalo 20-35 épocas; e por fim, que o pós-processamento em geral, melhora os resultados. Para o melhor caso (usando o MVSVM não linear sem PCA) para uma classificação com 13 pessoas, obteve-se 15,67% de erro e após o pós-processamento, 5,10% de erro.

keywords

Personal Identification and Autentification, Electroencefalogram, Brain, Visual Evoked Potencials.

abstract

The main objective of this thesis is to develop a personal identification system based on EEG signals. EEG signals are acquired through 22 electrodes placed on the scalp (20 to 2 for EEG and EOG), while the individual is subject to visual stimuli. The feasibility of this biometric system is tested with data from 16 individuals, implementing several methods of classification and signal processing.

The first chapter is an introduction to the concept of Biometric Identification and potential of this model for the near future. This chapter also indicates the objectives of the thesis.

In the second chapter, is presented some theoretical concepts, which are the basis of this thesis, for a better understanding of the work in question. They also described some related works that served as a starting point for the work.

In the third chapter, Experimental Procedure, is explained the methodology used to achieve the results. It is described all the processes that led to obtain the results, in the following chapter, from the identification of individual characteristics to the Classification.

In the fourth chapter is exposed the results obtained for the various classifiers discussed: with and without PCA; and linear and non-linear. The classifiers used were the MVSVM (Majority Voting Support Vector Machine) and OAASVM (One Against All Support Vector Machine).

In the fifth and final chapter (Conclusions), it was concluded that: the rejection of epochs with artefacts improves the final results; the PCA applied to the EEG signals available, contributes to the increase of the error in classification; classifiers in the nonlinear case are more effective than linear; the training and testing of the classifiers must be made with data from all carried out experiments to obtain the PEVs; the ideal number of times for testing a non-linear classifier is somewhere between the range 20-35 times; and finally that the post-processing in general, improves the results.

For the best case (using the nonlinear MVSVM without PCA) for classification with 13 people, it was obtained 15.67% of error and after post-processing, an error of 5.10%.

Índice

Capítulo 1

Introdução	19
1.1 Objectivos:	19

Capítulo 2

Materiais e Métodos	20
2.1 Conceitos teóricos.....	20
2.1.1 Potenciais evocados	23
2.1.2 Potenciais evocados visuais (PEV)	23
2.1.3 Potenciais evocados auditivos (PEA)	23
2.1.4 Potenciais evocados Somatossensitivos (PESS)	24
2.2 Tipos/algoritmos de classificação.....	24
2.2.1 k -Nearest Neighbors algorithm (k -NN).....	24
2.2.2 Redes neuronais	25
2.2.3 Support Vector Machine (SVM).....	25
2.3 Esquema base de um sistema de identificação	32
2.4 Estado da Arte.....	33

Capítulo 3

Realização experimental.....	35
3.1 Dados utilizados.....	35
3.1.1 Imagens utilizadas como estímulos.....	35
3.1.2 Aquisição de Dados	36
3.1.3 Registo do sinal EEG	37
3.2 EEGLAB.....	37
3.3 Identificação das características.....	39
3.3.1 Diagrama de blocos.....	40

3.3.2 Rejeição de épocas com artefactos.....	41
3.3.3 PCA.....	42
3.3.4 Filtragem da banda Gama (30-50Hz) e cálculo das energias dos canais	46
3.3.5 Normalização das energias.....	48
3.3.6 Formação das bases de dados.....	49
3.3.7 Formação das bases de dados, sem PCA	50
3.4 Classificação	51
3.5 Pós-Processamento	53
Capítulo 4	
Resultados Experimentais.....	57
4.1 Parâmetros do classificador SVM.....	56
4.2 Resultados com PCA	57
4.2.1 Rejeição das épocas com artefactos	57
4.2.2 Classificador MVSVM não linear.....	58
4.2.3 Classificador MVSVM linear	60
4.2.4 Classificador OAASVM linear	63
4.2.5 Classificador OAASVM não linear	65
4.3 Resultados sem PCA.....	67
4.3.1 Classificador MVSVM não linear.....	67
4.3.2 Classificador MVSVM linear	69
4.3.2 Classificador OAASVM linear	71
4.3.3 Classificador OAASVM não linear	73
4.4 Resultados finais.....	75
Capítulo 5	
Conclusões.....	79
Referências.....	81
Anexos.....	83

Índice de Figuras

Fig.2.1 Posição dos eléctrodos no crânio.....	21
Fig.2.2 Sinal EEG com 20 canais, mais os 2 canais EOG e 5 épocas.....	22
Fig.2.3 Diagrama do funcionamento simplificado do mecanismo de classificação.....	24
Fig.2.4 Escolha de Hiper-planos possíveis (a azul e vermelho), ou inválidos (verde), para divisão da <i>data</i>	26
Fig.2.5 Escolha do Hiper-plano em função da margem máxima.....	27
Fig.2.6 Divisão de um conjunto de dados bidimensional.....	28
Fig.2.7 Mudança de transformação de dados.....	31
Fig.2.8 Diagrama de blocos do sistema de identificação.....	32
Fig.3.1 Exemplo de estímulos resultantes da sobreposição entre as casas e as faces, com variação das expressões faciais: (A) Neutro; (B) Medo; (C) Nojo.....	36
Fig.3.2 Janela do EEGLAB.....	38
Fig.3.3 EEGLAB carregado com os dados do primeiro sujeito.....	38
Fig.3.4 Diagrama de blocos representativo do processo de extracção das características(vector de energias normalizadas) de um indivíduo.....	40
Fig.3.5 Gráfico do sinal EEG do sujeito 1 das primeiras 5 épocas.....	41
Fig.3.6 Gráfico da percentagem de <i>eigenvalues</i> em função dos <i>eigenvectors</i>	45
Fig.3.7 Gráfico do resultado da função <i>pwelch</i> para um canal de um determinado evento.....	47
Fig.3.8 Gráfico das energias dos canais de um determinado indivíduo, para um evento.....	48
Fig.3.9 gráfico das energias normalizadas na banda Gama por canal.....	49
Fig.3.10 Diagrama de blocos do processo total de identificação das características.....	49
Fig.3.11 Diagrama de blocos relativo à identificação das características sem recurso à PCA.....	50
Fig.3.12 Gráfico de uma classificação com 5classes.....	53
Fig.3.13 Gráfico de uma classificação e da correcção da mesma usando a primeira iteração do pós-processamento.....	54
Fig.3.14 Gráficos do resultado de uma classificação e de três iterações do pós-processamento...	55
Fig.4.1 Gráfico dos resultados de classificação e pós-processamento, para os sinais com (sinal1) e sem artefactos (sinal2).....	57
Fig.4.2 Gráfico dos resultados da classificação do MVSVM não linear.....	59
Fig.4.3 Gráfico da variação do valor do erro da classificação alternada, desde 7 a 120 épocas....	60

Fig.4.4 Gráfico dos resultados da classificação do MVSVM linear.....	62
Fig.4.5 Gráfico da variação do erro da classificação alternada, desde 7 a 120 épocas.....	62
Fig.4.6 Gráfico dos resultados da classificação do OAASVM linear.....	63
Fig.4.7 Gráfico da variação do erro da classificação OAASVM linear, desde 7 a 120 épocas.....	64
Fig.4.8 Gráfico dos resultados da classificação do OAASVM não linear.....	65
Fig.4.9 Gráfico da variação do erro da classificação OAASVM não linear, desde 7 a 55 épocas...	66
Fig.4.10 Gráfico dos resultados da classificação do MVSVM não linear.....	68
Fig.4.11 Gráfico da variação do erro da classificação MVSVM não linear, desde 7 a 120 épocas..	68
Fig.4.12 Gráfico dos resultados da classificação do MVSVM linear.....	71
Fig.4.13 Gráfico da variação do erro da classificação MVSVM linear, desde 7 a 120 épocas.....	71
Fig.4.14 Gráfico dos resultados da classificação do OAASVM linear.....	72
Fig.4.15 Gráfico da variação do erro da classificação OAASVM linear, desde 7 a 120 épocas.....	72
Fig.4.16 Gráfico dos resultados da classificação do OAASVM não linear.....	73
Fig.4.17 Gráfico da variação do erro da classificação OAASVM não linear, desde 7 a 120 épocas.....	74
Fig.4.18 Gráfico dos resultados dos classificadores com o uso da PCA.....	75
Fig.4.19 Gráfico dos resultados dos classificadores sem uso da PCA.....	75

Introdução

A análise biométrica, tem vindo a ser cada vez mais, uma ferramenta de identificação pessoal ao serviço das mais diversas instituições, governamentais e não governamentais. Num mundo onde, a produtividade, segurança, a luta contra o terrorismo e criminalidade é cada vez mais sinónimo de necessidade e desenvolvimento, torna-se imperativo o investimento em recursos e tecnologias que honrem estas premissas.

O caso mais comum e mais popular, é o caso da identificação por impressão digital, utilizada nos antigos BI (Bilhete de Identidade). O processo baseia-se no facto de que cada indivíduo possui uma impressão digital característica e única, principio este que é a própria definição e fundamento base da análise biométrica, isto é, cada indivíduo possui características fisiológicas que o caracterizam como único. No entanto, este processo (impressão digital) tem vindo a ser questionado [1] e outros métodos têm sido explorados para complementarem este processo de autentificação.

Outros métodos bastante difundidos são por exemplo a autentificação através do DNA, fisiologia da face, palma das mãos [2], e a própria íris, etc. Existem já alguns países que implementaram passaportes biométricos, como é o caso da Alemanha e futuramente outros países da união europeia, Portugal inclusive. Algumas outras abordagens estão a ser idealizadas, algumas delas levantando questões éticas, como é o caso do implante de chips no corpo humano.

Neste trabalho vamos abordar a autentificação biométrica baseada na actividade cerebral dos indivíduos, mais concretamente a resposta a PEV (potenciais evocados visuais), que de uma forma muito simplificada, é a resposta do processamento visual de determinadas imagens. Relativamente a alguns dos métodos apresentados anteriormente, este destaca-se em vários factores, nomeadamente no facto de ser impossível reproduzir os padrões cerebrais de determinado sujeito, devido à sua singularidade. A derradeira questão é que, com o conhecimento que hoje possuímos no domínio do cérebro humano, não nos é possível “medir”, “observar” e copiar a forma como cada um de nós “vê o mundo”.

1.1 Objectivos:

O objectivo principal desta tese é desenvolver um sistema de identificação pessoal baseado em sinais EEG. Os sinais de EEG são adquiridos através de 22 eléctrodos colocados no escalpe (20 para EEG e 2 para EOG), enquanto o indivíduo for sujeito a estímulos visuais. A viabilidade deste sistema biométrico é testada com os dados de 16 indivíduos implementando vários métodos de classificação e processamento de sinal.

Materiais e Métodos

A biometria (do Grego, bio - vida, metron - medida), como o nome indica, é a ciência que estuda as características fisiológicas e comportamentais dos seres vivos. Nos últimos anos, a biometria tem vindo a ter uma crescente importância no nosso quotidiano, e conseqüentemente, cada vez mais existem métodos biométricos para colmatar as necessidades do homem do século XXI.

Existem já exemplos da utilização destas novas tecnologias ao serviço do cidadão do mundo industrializado, desde o controlo de passageiros nos aeroportos através da geometria da mão, até ao próprio voto biométrico que possivelmente, num futuro não muito distante, fará parte do nosso quotidiano. Significa portanto que a confiança nestas inovações e fiabilidade das mesmas vão torna-las responsáveis por garantir algo como por exemplo o dever e direito máximo como cidadãos democráticos.

A biometria divide os métodos de identificação em dois grupos, o grupo da identificação biométrica através de processos fisiológicos e através de processos comportamentais.

O campo de investigação biométrica tem vindo a crescer ao longo dos anos. A necessidade de nos conhecermos, identificar e distinguir com certeza um determinado indivíduo de outros, leva a uma busca pelo conhecimento pormenorizado da fisionomia e todos os processos intelectuais que nos tornam únicos. Exemplos disso são, os já referidos métodos de identificação biométrica, mas este trabalho centra-se num tipo de identificação muito particular, nomeadamente a identificação através da resposta intelectual de um indivíduo a um estímulo visual. Este processo implica, de certa forma, um conhecimento da maneira de pensar do cérebro humano e conhecer que tipos de comportamentos são desencadeados quando uma imagem é colocada perante uma pessoa.

2.1 Conceitos teóricos

Para melhor se compreender o processo de identificação baseado nos sinais EEG, é necessário conhecer as características do sinal a estudar.

Neste trabalho, a actividade eléctrica do cérebro é medida através de 22 eléctrodos (pequenas placas metálicas), que estão colocadas sobre a pele do crânio dos indivíduos, em posições específicas.

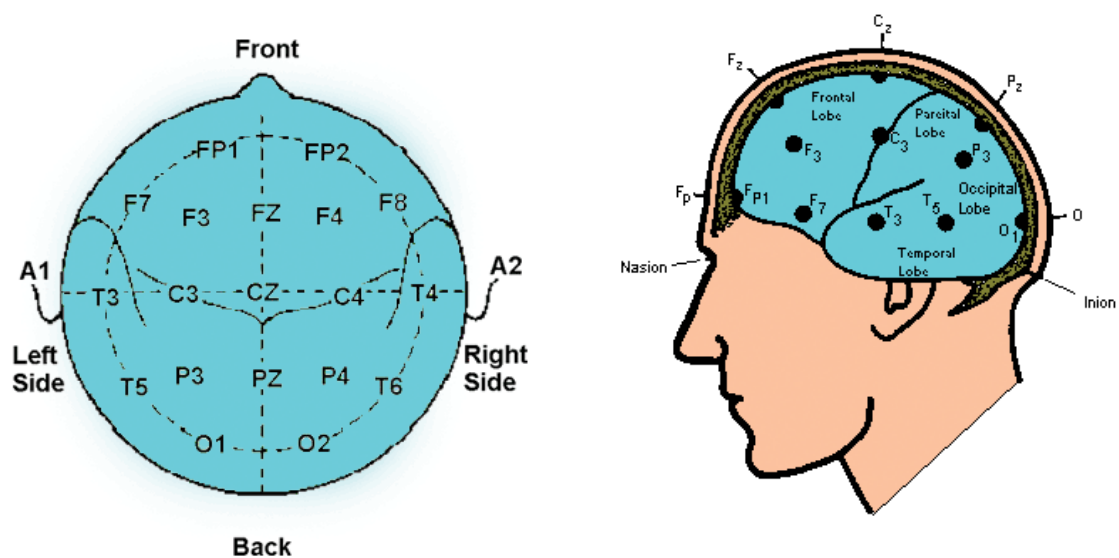


Fig.2.1 Posição dos eléctrodos no crânio

Um dos sistemas mais usados de colocação de eléctrodos no crânio, é o sistema internacional 10-20 (fig.2.1). O sistema é baseado numa relação entre a posição dos eléctrodos no crânio e na área do cérebro sobre o qual estão localizados. Os lobos cerebrais estão caracterizados com letras, como por exemplo: o “F” significa Frontal; o “P”, Parietal; o “T”, Temporal; e “O” Occipital.

Os números pares correspondem à parte direita do hemisfério e os ímpares à esquerda, a letra “Z” é referente à linha central entre os dois hemisférios (esquerdo e direito). Quanto menor for o número, menor a distância em relação à linha central. Os números ou as letras seguintes representam, juntamente com a primeira letra a posição específica no hemisfério cortical.

Existem por vezes, uns eléctrodos extra, que são posicionados entre os que foram anteriormente referidos o que implica necessariamente um novo sistema de nomenclatura, a Nomenclatura de Combinação Modificada (MCN). [16]

O sinal EEG é proveniente da diferença de potencial eléctrico entre os diferentes eléctrodos, amplificado e filtrado, antes de ser registado digitalmente. Estas ondas cerebrais geralmente têm uma amplitude entre 0 e 200mV e são classificadas segundo a sua frequência (varia entre 0.5 e os 50Hz), localização, amplitude e simetria em relação à linha média que divide os dois hemisférios (esquerdo e direito).

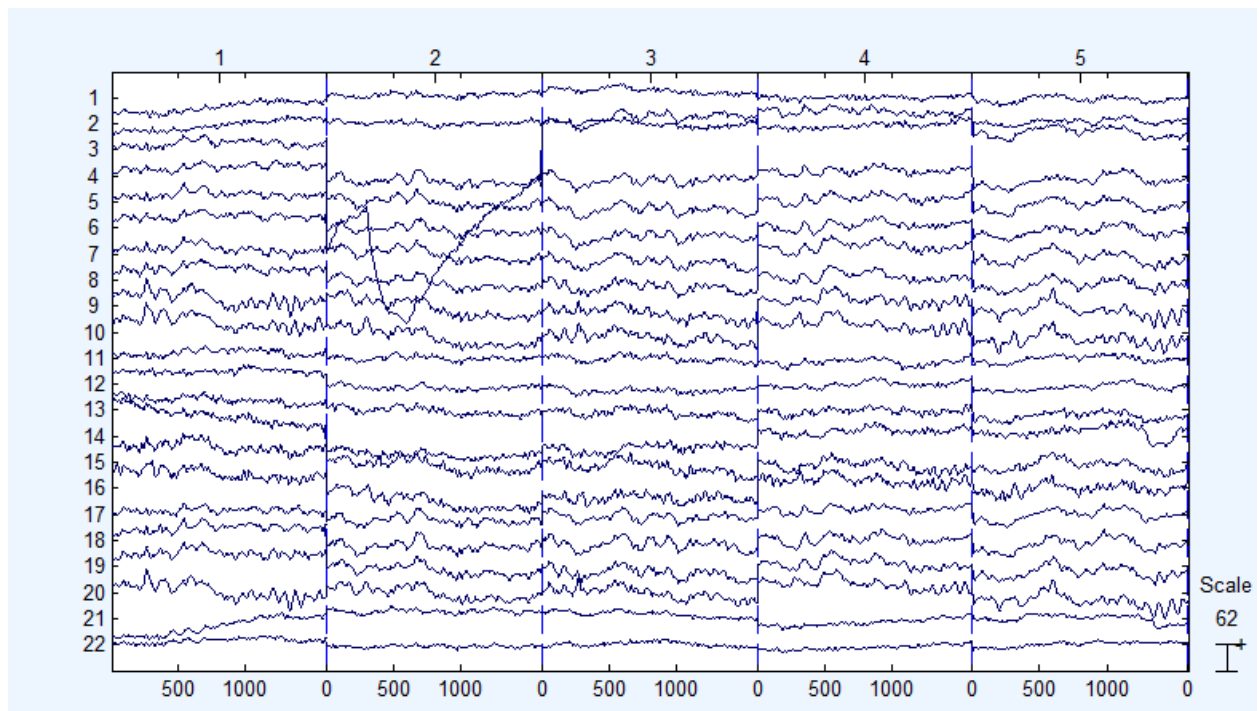


Fig.2.2 Sinal EEG com 20 canais, mais os 2 canais EOG e 5 épocas

Quanto a classificação das ondas cerebrais, no que diz respeito às bandas de frequência, temos o espectro dividido em cinco. Esta divisão não é muito consensual no que diz respeito às frequências limite das bandas, no entanto as diferenças não são muito significativas.

A banda mais baixa é a banda Delta que abrange as frequências de aproximadamente 0.5 a 4Hz. Esta banda encontra-se mais activa durante o sono, estado relacionada essencialmente com o subconsciente.

Depois da banda Delta, vem a banda Teta, que comporta as frequências desde 4 a 8Hz. Estas ondas são predominantes durante o sono leve.

A banda Alfa possui as frequências de 8 a 13Hz e está relacionada com a atenção e o estado de expectativa. Existem vários trabalhos relacionados com esta banda, em que, para melhor serem registadas estas ondas, os sujeitos são expostos a um estímulo visual. Os sujeitos processam-no de olhos fechados, não sendo registadas as respostas relativas ao estímulo, mas sim ao processamento do mesmo. São registadas as ondas desfasadas do instante em que foi apresentada a imagem inicial.

Às frequências de 13 a 30Hz, denominamos de banda Beta. Esta banda normalmente apresenta uma classificação que subdivide a gama de 13 a 30Hz em três bandas: Beta1 (12-15Hz), Beta2 (15-18Hz) e Beta3 (18-30Hz). De um modo geral a banda Beta está relacionada com actividades neuronais desde um estado de relaxamento e concentração (característico da Beta1) e o estado de alerta, consciência activa e agitação (Beta3).

Finalmente, entre 30 e 50Hz, a banda Gama. Esta banda foi a utilizada nesta tese por ser a responsável por actividades cerebrais de alto nível. Exemplos disso são: a memória; processamento de estímulos visuais e outros processos intelectuais próprios de

um estado de pela consciência, como é o caso dos indivíduos, de onde foram retirados os dados para este trabalho.

2.1.1 Potenciais evocados

Como foi referido anteriormente foram usados potenciais evocados visuais (PEV), como base de estudo, neste trabalho. Para melhor se compreender a utilidade destes sinais na obtenção de resultados, que vão de acordo com os objectivos propostos, torna-se relevante perceber exactamente o que são potenciais evocados e fundamentalmente, o porquê da utilização destes sinais na identificação biométrica.

Um potencial evocado é uma resposta do sistema nervoso central e periférico, a estímulos específicos. Esta propriedade do cérebro é usada para monitorização da actividade cerebral, para, por exemplo, detecção de doenças neurológicas, durante uma cirurgia, ou para estimular determinadas áreas do córtex cerebral que são responsáveis pelo processamento de determinadas funções, intelectuais ou motoras. Existem três tipos principais de potenciais evocados, cuja utilização é mais frequente.

2.1.2 Potenciais evocados visuais (PEV)

Este tipo de potenciais, são causados pelo estímulo sensorial do campo visual de um indivíduo. Normalmente são imagens simples, geralmente a preto e branco ou flashes, colocadas diante do sujeito durante o EEG, com um tempo de exposição relativamente curto.

São usados para detecção de problemas visuais, como cegueira ou problemas relacionados com o nervo óptico. Quando não existe registo de resposta neuronal característica de um PEV, significa que não existe comunicação entre os olhos e o cérebro.

As propriedades do PEV, fazem destes uma ferramenta bastante fiável, para identificação biométrica.

Trabalhos semelhantes, [6][12], como o apresentado nesta tese, mostram que dados provenientes de PEVs, são bastantes razoáveis, para se atingir bons resultados. Isto deve-se ao facto de os PEVs potenciarem respostas, que são específicos de cada indivíduo e não se alterarem muito significativamente para a mesma pessoa mesmo quando são usados estímulos visuais diferentes. Foram estas características que justificaram a utilização destes dados.

2.1.3 Potenciais evocados auditivos (PEA)

Os PEAs são resultantes de estímulos sonoros, usados para detecção de problemas auditivos, testando a conexão entre o nervo acústico, responsável pela condução da informação até ao cérebro, e o sistema nervoso. Também são usados para detectar morte cerebral, deficiências de recém-nascidos, avaliar processos de linguagem, etc.

2.1.4 Potenciais evocados Somatossensitivos (PESS)

Este tipo de potenciais evocados são usados para testar a transmissão de sinal nervoso entre os membros até ao cérebro. A sua utilização está relacionada com a detecção de lesões nas raízes nervosas, espinal medula e encéfalo. Estes testes são efectuados com recurso a pequenos choques de corrente eléctrica, localizadas num nervo específico dos membros.

2.2 Tipos/algoritmos de classificação

Classificação é a função de atribuir objectos a uma das diferentes, predefinidas classes ou categorias. Existem vários exemplos do uso de classificadores no nosso quotidiano, exemplos disso são classificadores que detectam mensagens *spam* que chegam aos emails.

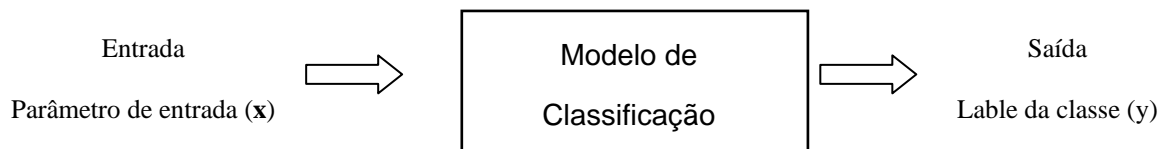


Fig2.3 Diagrama do funcionamento simplificado do mecanismo de classificação

O diagrama da figura 2.3 representa, muito simplificada, a mecânica do processo de classificação. Como entrada do modelo, tem-se x , a informação que se pretende classificar. Este objecto de entrada é caracterizado como (x,y) , onde x é o atributo a classificar e y é a classe correspondente, isto é, a designação da classe a que pertence x . À saída do diagrama tem-se o *lable* y correspondente ao resultado da decisão do classificador, mais concretamente a classe a que o modelo atribuiu x .

Classificação, por definição é a tarefa de encontrar uma dada função f , que seja capaz de mapear/atribuir a cada x uma classe y predefinida.

2.2.1 k -Nearest Neighbors algorithm (k -NN)

K -NN é um método de classificação baseado na distância entre objectos, isto é, a classificação é feita com base na menor distância (exemplo disso é a distância Euclidiana) entre os objectos. A classificação pode ser feita para $k=1$, $k=3$, $k=5$, etc., isto é, k é sempre um número ímpar que representa o número de "vizinhos", usados para ser efectuada a classificação numa determinada classe ou grupo. A classe que tiver mais vizinhos, do objecto a classificar, é a classe a que o objecto pertence.

2.2.2 Redes neuronais

Redes neuronais são, como o nome indica, redes de neurónios, o exemplo mais tradicional da aplicação do termo, são as existentes nos cérebros humanos.

Actualmente o termo redes neuronais divide-se em:

- i) redes neuronais biológicas (RNB) que, se referem às estruturas que formam os canais de comunicação entre os neurónios constituintes nos sistemas nervosos;
- ii) redes neuronais artificiais (RNA), que corresponde a uma técnica matemática.

As RNA são constituídas por camadas de funções matemáticas, chamadas neurónios artificiais, que se aproximam em termos de função aos neurónios biológicos. Apesar das RNA terem sido inspiradas pelas RNB, estes são modelos matemáticos com muitas aplicações.

Na prática, as RNA, são um sistema de computação paralela, com um conjunto de pequenas tarefas de processamento simples, que quando interligadas entre si pretendem efectuar uma tarefa específica.

Obviamente que as RNA são uma aproximação grosseira quando comparadas com as RNB, em termos de complexidade e eficiência. Apesar disto as redes neuronais são muito eficientes em termos de processamento: têm a capacidade de ser treinadas; são tolerantes ao ruído e ao erro que existem abundantemente em dados, entre outras vantagens que decorrem da utilização desta ferramenta de classificação.

As RNA são uma ferramenta muito utilizada nas áreas de classificação e reconhecimento de padrões. As suas propriedades de adaptação, aprendizagem e processamento em paralelo, fazem das RNA uma ferramenta poderosa para reconhecimento de padrões, tais como os padrões específicos de cada individuo presentes num PEV.

Existem vários algoritmos que visam esta minimização do erro, tais como:

- o algoritmo de *back propagation*, que é também um dos mais usados;
- o *feed-forward* ANN onde não existe realimentação no processo de processamento elementar por camadas, isto é, após uma camada de processamento, o resultado não volta a ser processado na mesma camada ou por uma camada anterior, dando origem a um fluxo unidireccional de processamento; entre outros.

2.2.3 Support Vector Machine (SVM)

O classificador SVM, tem vindo a tornar-se bastante popular e cada vez mais, alvo de discussão. Os princípios deste método residem na teoria de aprendizagem estatística. Esta aprendizagem é conseguida através de um treino a que o classificador é sujeito, usando dados iguais ou muito parecidos com os quais vai efectuar a classificação.

Depois deste treino é que o classificador está em condições de decidir, em função da informação que recolheu anteriormente.

Tem sido utilizado em diversas aplicações práticas, exemplo disso são: a sua utilização no reconhecimento digital de caligrafia; categorização de texto, entre outros. No entanto, este classificador destaca-se sobretudo no campo dos dados hiper-dimensionais, uma vez que evita o problema de dimensionamento, característico de classificadores que lidam com dados com elevado número de dimensões.

Para melhor se compreender o conceito do SVM, primeiro vai fazer-se referência ao conceito de margem máxima do hiper-plano e posteriormente apresenta-se como é aplicado um SVM linear e depois SVM aplicado a dados com características não lineares.

Margem Máxima do Hiper-plano

Considere-se um conjunto dados, correspondentes a duas classes. O objectivo do classificador é separar os dados em duas classes, uma vez que o classificador SVM é binário.

No conjunto de dados representado na figura 2.4, pode-se identificar duas classes, as que apresentam os pontos preenchidos a preto e as que são círculos.

Estes conjuntos são linearmente separável, uma vez que é possível, recorrendo a um hiper-plano com dimensão $p-1$ dimensões, em que p é a dimensão dos dados a classificar, para separar as classes. No entanto existe infinitas possibilidades para dividir o plano dos dados da Fig2.3, como por exemplo os planos H_1 e H_2 . No caso do plano H_3 , facilmente se visualiza que não é uma correcta escolha para divisão do *dataset*.

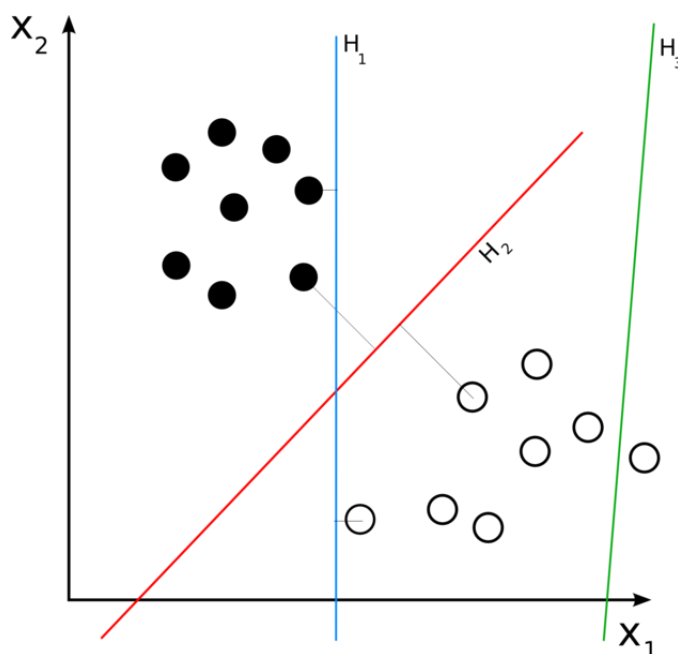


Fig2.4 Escolha de Hiper-planos possíveis (a azul e vermelho), ou inválidos (verde), para divisão da *data* [18]

Apesar do facto de no processo de treino, o classificador ter um erro nulo, obviamente não se pode garantir que o hiper-plano escolhido tenha o mesmo

desempenho com dados desconhecidos, aquando da fase de treino. Por esta razão é preciso escolher o hiper-plano que melhor representa a divisão entre as diferentes classes. Este critério de decisão assenta no princípio, de que se deve escolher o hiper-plano que apresenta uma maior margem de divisão, de modo a que os dois conjuntos de dados se encontrem o mais afastados possível, em relação ao plano de separação.

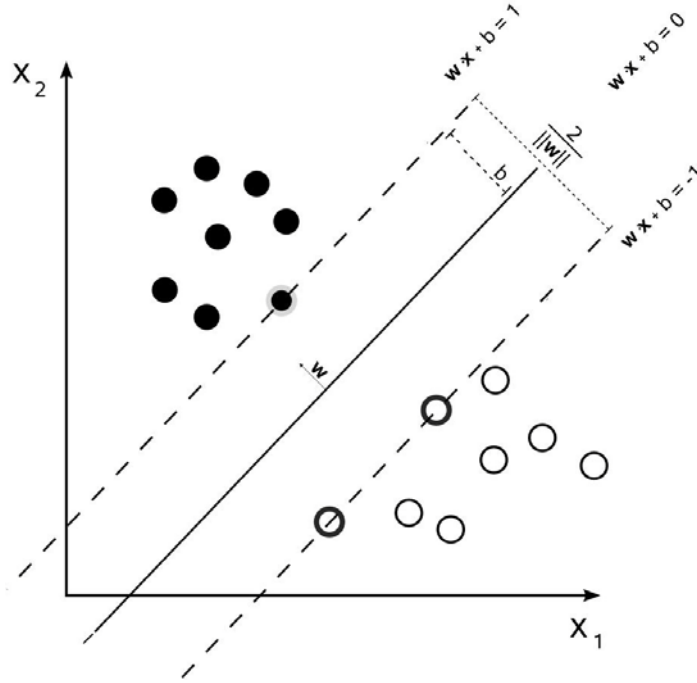


Fig2.5 Escolha do Hiper-plano em função da margem máxima [18]

Essa margem máxima é determinada através da colocação de dois planos paralelos (b_{i1} e b_{i2}) ao plano central (B_i), deslocando estes dois planos, b_{i1} e b_{i2} até intersectarem um ponto de cada uma das classes. Esta distância entre os b_{i1} e b_{i2} , é que define a margem do classificador. Na fase de treino, o que o classificador faz é escolher o hiper-plano B_i que possui a maior margem de decisão.

A escolha do plano em função da sua margem para classificadores lineares, assenta no princípio de Minimização do Risco Estrutural (SRM, do inglês Structural Risk Minimization). Este princípio devolve um valor máximo de erro (R), para um classificador em termos do seu erro de treino (R_e), número de amostras de treino (N) e do modelo de capacidade, também conhecido com capacidade (h). A generalização do erro do classificador, com uma probabilidade de $1-\eta$ é no pior dos casos [18]:

$$R \leq R_e + \varphi\left(\frac{h}{N}, \frac{\log(\eta)}{N}\right), \quad (2.1)$$

onde φ é uma função que representa o crescimento monótono da capacidade h .

A capacidade de um modelo linear, está inversamente relacionada com as suas margens. Os modelos com grande capacidade tem pequenas margens, pois devido à sua flexibilidade podem adaptar-se a um maior número de modelos de dados de treino, contrariamente aos modelos com grandes margens. Apesar destas características, para se criar um classificador linear com um erro genérico mínimo, devemos usar a maior

margem possível, pois de acordo com o princípio SRM, quanto maior for a margem do classificador menor o valor de R .

Classificador SVM linear

(Caso em que os dados são separáveis)

Se considerar-se uma classificação binária consistente em N exemplos de treinos, onde cada exemplo de treino é representado por \mathbf{x}_i, y_i (com $i = 1, 2, \dots, N$), em que $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$ e $y_i \in \{-1, 1\}$, representa a designação da classe. Deste modo, pode-se escrever a equação de decisão de margem do classificador como:

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \quad (2.2)$$

onde \mathbf{w} e b são parâmetros do modelo que definem a localização do hiper-plano.

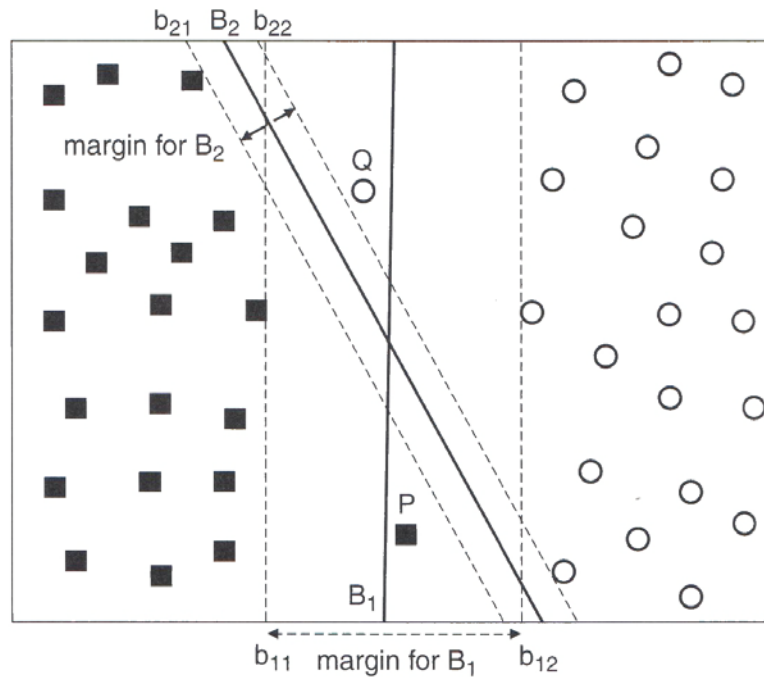


Fig2.6 Divisão de um conjunto de dados bidimensional[18]

Considerando ainda um modelo de treino bidimensional, como o da Fig2.6, onde se encontra representado o hiper-plano que bissecta os exemplos de treino em duas classes com a linha sólida e com duas linhas a tracejado observam-se as margens do classificador. Neste exemplo, os dois vectores correspondentes aos dois pontos (\mathbf{x}_a e \mathbf{x}_b) que se encontram sobre a linha do hiper-plano B_1 ,

$$\mathbf{w} \cdot \mathbf{x}_a + b = 0,$$

$$\mathbf{w} \cdot \mathbf{x}_b + b = 0$$

Ao se subtrair estas duas equações temos:

$$\mathbf{w} \cdot (\mathbf{x}_b - \mathbf{x}_a) + b = 0 \quad (2.3)$$

$\mathbf{x}_b - \mathbf{x}_a$ é um vector paralelo ao hiper-plano, com direcção de \mathbf{x}_a para \mathbf{x}_b . Como o produto vectorial é zero, \mathbf{w} é perpendicular à decisão.

Para qualquer um dos círculos situados à direita do hiper-plano, atribuímos

$$\mathbf{w} \cdot \mathbf{x}_s + b = k, \quad (2.4)$$

do mesmo modo, para os quadrados à esquerda do hiper-plano,

$$\mathbf{w} \cdot \mathbf{x}_c + b = k', \quad (2.5)$$

Pode-se então afirmar que qualquer ponto (\mathbf{x}_s) que se situa à direita da linha de divisão possui um valor de k maior que zero, do mesmo modo, qualquer ponto (\mathbf{x}_c) à esquerda da divisória tem um valor de k' negativo. Se classificarmos os quadrados com $y = -1$ e os círculos com $y = +1$, para qualquer exemplo de teste, temos:

$$y = \begin{cases} 1, & \text{se } \mathbf{w} \cdot \mathbf{z} + b > 0; \\ -1, & \text{se } \mathbf{w} \cdot \mathbf{z} + b < 0. \end{cases} \quad (2.6)$$

Margem de um classificador linear

No caso da Fig2.6, segundo a equação 2.4, pode-se afirmar que, uma vez que os círculos se encontram à direita da linha de divisão, possuem um valor de k positivo, e contrariamente, os quadrados possuem um valor negativo. Estes valores podem ser escalados de modo a que os círculos e quadrados que se encontram sobre as linhas de margem, paralelas à divisão (*bi1* e *bi2*), possuam o valor de $+1$ e -1 , respectivamente. Este ajuste da escala é feito através de ajustes dos parâmetros \mathbf{w} e b . Desta forma:

$$b_{i1} : \mathbf{w} \cdot \mathbf{x}_a + b = 1 \quad (2.7)$$

$$b_{i2} : \mathbf{w} \cdot \mathbf{x}_b + b = -1 \quad (2.8)$$

Como a margem de decisão é dada pela distância (d) entre os hiper-planos *bi1* e *bi2*, basta subtrair-se a equação 2.7 à equação 2.8 e temos:

$$\mathbf{w}(\mathbf{x}_a - \mathbf{x}_b) = 2 \quad (2.9)$$

$$d = \frac{2}{\|\mathbf{w}\|}. \quad (2.10)$$

Para o processo de treino, o que o classificador faz é tentar colocar os planos *bi1* e *bi2* à máxima distância. Encontrar este valor máximo de d é o mesmo que minimizar a seguinte função:

$$f(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2}. \quad (2.11)$$

O problema da tarefa de treino do SVM, esta relacionado com o seguinte problema:

$$\min_{\mathbf{w}} = \frac{\|\mathbf{w}\|^2}{2},$$

$$\text{para } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, N.$$

Para se resolver este problema aplica-se o método de Lagrange:

$$Lp = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^N \lambda_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1), \quad (2.12)$$

onde os parâmetros λ_i são chamados de multiplicadores de Lagrange. Para se determinar estes multiplicadores, é necessário derivar parcialmente a equação 2.12, em ordem a \mathbf{w} e b .

Em seguida recorre-se à transformação da inequação, apresentada como problema, em duas condições, isto só é possível se os multiplicadores de Lagrange forem não negativos. Chegou-se então às condições Karush-Kuhn-Tucker (KKT):

$$\lambda_i \geq 0, \quad (2.13)$$

$$\lambda_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] = 0. \quad (2.14)$$

Substituindo na equação 2.12 o resultado das derivadas parciais, chegamos aos valores dos multiplicadores de Lagrange. Com os valores de λ_i encontrados, para determinar \mathbf{w} e b basta usar a equação 2.14 e a derivada parcial em ordem a \mathbf{w} da equação 2.14:

$$\mathbf{w} = \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i \quad (2.15)$$

$$b = \frac{1}{y_i} - \mathbf{w} \cdot \mathbf{x}_i \quad (2.16)$$

Por vezes, há casos em que os dados não são totalmente separáveis, ou por motivos de eficiência do classificador, é preferível introduzir um pequeno erro no processo de treino, de modo a se utilizar o hiper-plano com as margens máximas. Este método de resolução é intitulado de *soft margin*, e assenta no facto de se introduzir uma variável constante de erro (ξ), para cada exemplo de treino, às equações 2.7 e 2.8. Esta introdução de erro tem de ser tida em conta na determinação da distância mínima entre os planos que definem a margem do classificador. Deste modo, a função que será alvo de minimização é:

$$f(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2} + C \left(\sum_{i=1}^N \xi_i \right)^k. \quad (2.17)$$

Nesta função, N representa o número de exemplos de treino que ficam dentro das margens do classificador e que tem de ser tidas em conta. C e k são parâmetros que se tem de definir e representam a penalização da não classificação dos elementos de treino. Quanto ao método de Lagrange, o método é idêntico ao caso linearmente separável, no entanto existem algumas condições diferentes, a que os multiplicadores de Lagrange (λ_i) têm de obedecer. Para o caso em que os dados são linearmente separáveis, λ_i tem de ser maior ou igual a zero, mas para o caso não separáveis, os valores de λ_i não podem exceder o valor de C . Resumindo, o caso linearmente separável $\lambda_i \geq 0$ e para o não linearmente separável $0 \leq \lambda_i \leq C$.

O valor de C , na prática é determinado dentro de intervalo por uma função cujo objectivo é encontrar o valor ideal de C e σ cujo valor é necessário para a classificação não linear, para solucionar a função de Kernel.

Classificador SVM não linear

Por vezes pretende-se classificar, conjuntos de dados que não são linearmente separáveis, isto é, que a superfície de treino não é um hiper-plano. Nesses casos, a abordagem a tomar é a de transformar as coordenadas espaciais x (que se trata do espaço original dos dados) num novo espaço $\phi(x)$ (espaço novo das características), onde seja possível usar limites de separação lineares. Depois de feita esta transformação, o método de resolução é semelhante ao utilizado para encontrar o melhor hiper-plano, dos classificadores lineares.

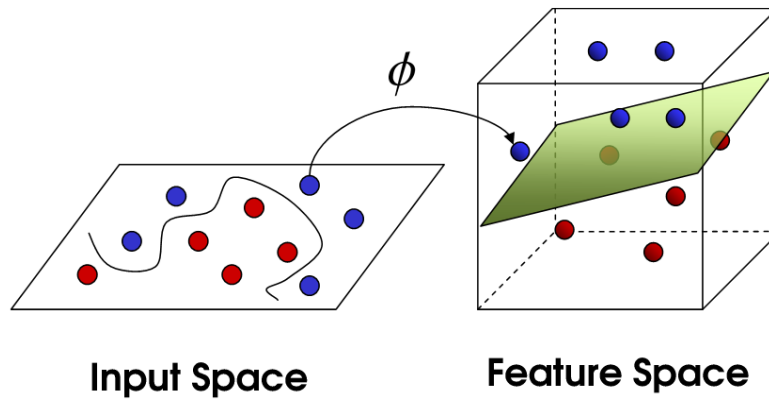


Fig2.7 Mudança de transformação de dados

A figura 2.7 ilustra um exemplo da transformação referida anteriormente, de modo a se poder recorrer a um hiper-plano, para se dividir linearmente as duas classes de dados.

Através do teorema de Lagrange, chega-se a uma equação que contém o produto interno entre $\phi(x_i)$ e $\phi(x_j)$. Este produto interno geralmente apresenta-se como uma medida de semelhança entre dois vectores. O produto interno é solucionado através de um método de kernel e o mais usual é o seguinte:

$$K(x_i, x_j) = e^{-\|x_i - x_j\|^2 / (2\sigma^2)} \quad (2.18)$$

A grande vantagem deste método encontra-se nesta propriedade de se conseguir, apesar da mudança de *Input Space* para *Feature Space*, determinar o produto interno $\phi(x_i) \cdot \phi(x_j)$ dos dados no *Input space*.

Características fundamentais do SVM

Este método de classificação tem vindo a se tornar cada vez mais popular, isto deve-se a algumas das suas características, que o evidenciam em relação a outras metodologias.

Uma dessas características, é o facto de o problema do processo de aprendizagem do SVM, estar unicamente relacionado com a complexidade matemática que reside no método de minimização da equação 2.11. Para se solucionar esta dificuldade, existem vários algoritmos bastante eficientes disponíveis, acabando por solucionar o principal problema. Comparativamente, pode-se afirmar-se que este classificador comporta certas vantagens pois, o problema de outros métodos, como por exemplo as redes neuronais, é que por vezes, tenderem a apresentar soluções locais e não generalizadas a todo o universo de dados.

Outra das características, é o caso de a classificação do SVM, no processo de maximização da margem de decisão, recorrer à função Kernel para classificação não linear.

Apesar de os métodos de classificação apresentados serem para classificadores binários (classificação em duas classes), existem outros métodos que tornam o SVM num classificador multi-classe.

2.3 Esquema base de um sistema de identificação

Um sistema de identificação é composto por um conjunto de processos, que vão deste do momento do estímulo até ao resultado da classificação. No diagrama de blocos da figura 2.8, encontra-se representado, nos seus blocos fundamentais, o sistema de identificação utilizado nesta tese.



Fig2.8 Diagrama de blocos do sistema de identificação

O sistema é constituído por cinco blocos essenciais. Para se proceder à identificação. Numa primeira fase é colocado um estímulo (no caso desta tese visual) e imediatamente é registado o sinal EEG. Estes PE (PEV nesta tese) são registados e submetidos a um conjunto de procedimentos e filtragem, antes de se retirarem as características individuais. Estas características são o objecto de classificação, na fase final deste sistema.

2.4 Estado da Arte

Alguns trabalhos têm sido feitos no campo da análise de sinais provenientes de EEG. Os mais comuns têm o propósito de diagnosticar deficiências ou doenças, exemplo disso é a identificação de problemas no sistema nervoso, como a detecção de epilepsia, que de certa forma se destacam devido a elevados valores de amplitude das ondas cerebrais.

Na área de psicologia também foi feita alguma pesquisa a propósito da maneira única, como cada sujeito vê, sente e processa a realidade envolvente. Os dados utilizados neste trabalho são provenientes de um trabalho realizado nessa área e com o fundamento de comprovar a singularidade da resposta neuronal a estímulos visuais específicos.

Existem ainda trabalhos relacionados com a identificação de determinados padrões neuronais em indivíduos alcoólicos [15], através de uma análise espectral, recorrendo à decomposição do sinal EEG em gamas de frequência e estudando as diferenças existentes nos valores de energia dessas bandas, entre indivíduos alcoólicos e não alcoólicos.

Aparecem também outros estudos e trabalhos, que não têm ligação com processos de diagnóstico, exemplo disso são os trabalhos na área de robótica e informática, onde já existem mecanismos que permitem uma interface com os computadores através de sinais cerebrais. Numa aplicação directa desta tecnologia, existem projectos de investigação, que visam a utilização destes mecanismos de interface através de sinais cerebrais usando potenciais evocados, para auxiliar indivíduos com deficiências motoras. [14]

No campo da identificação biométrica recorrendo a sinais provenientes de EEG, *Poulos et al.*[4] foi o primeiro a propor um modelo para identificação. O método é baseado num algoritmo auto regressivo (AR), aplicado à banda alfa (8-12Hz) do sinal EEG. Usando os parâmetros do modelo AR como *features* (características) que formam os vectores de entrada para o modelo de identificação. Para classificação de um indivíduo, como sendo diferente de todos os outros indivíduos, num universo limitado, foi usada a quantificação vectorial linear de Kohonen, com resultados na ordem dos 72-80% de sucesso.

Posteriormente, *Paranjpe et al.*[5] seguiu também um método AR com uma análise discriminatória, para a identificação de indivíduos, obtendo uma classificação com resultados que oscilavam entre 49-85%. É de referir que estes dados foram registados, não através de PEVs mas enquanto os indivíduos se encontravam a descansar com os olhos fechados e posteriormente, com os olhos abertos. Neste trabalho foram usados apenas oito eléctrodos, correspondentes aos canais frontais (F7, F8), parietais (P3, P4) e temporais (T3, T4, T5 e T6). Para cada sujeito foram utilizadas oito épocas com 1024 amostras, amostradas a 120Hz. O processamento dos dados comporta duas fases. Na primeira fase, é calculada a variância/co-variância de todas as variáveis. Na segunda fase é calculada a variância/co-variância, mas desta feita para os dados de cada classe. Depois é maximizada a variância entre as classes e minimizada para os dados dentro das classes.

R. Palaniappan [6], mais recentemente, apresentou outra metodologia para esta problemática, que se provou bastante eficaz. O autor recorreu-se de PEV (potenciais evocados visuais), para a identificação. Filtrou por intermédio de um filtro digital *buttherworth* de fase zero, apenas a banda gama do sinal EEG (30-50Hz). As energias espectrais extraídas dos 61 canais usados (3 dos 64 canais são de referência), depois de normalizadas com as energias de todas as bandas de frequência do EEG, constitui-se, um vector das características de cada indivíduo (*feature vector*).

O uso da banda gama é justificado pelo facto de ter sido usada com sucesso, na classificação de indivíduos alcoólicos e não alcoólicos [7] e essencialmente porque tem sido demonstrado que esta gama está relacionada com funções de alto nível do cérebro, nomeadamente a percepção e memória [8-10]. Como os potenciais visuais evocados procuram estimular, entre outras, estas funções intelectuais, pensa-se que esta gama é idealizada para identificação pessoal.

Para classificação, Palaniappan usou, inicialmente, um método de *fuzzy classification*, e *backpropagation neural network* (BP NN), em vez da classificação paramétrica de Paranjpe. O uso do classificador BP NN, foi justificado por não necessitar de elevados tempos de treino para obter resultados aceitáveis, contrariamente a outros métodos de NN, que regra geral são difíceis de treinar.

R. Planiappan e K.V.R Ravi, publicaram posteriormente outros trabalhos, abordagens semelhantes, fazendo-se algumas alterações nomeadamente na fase de classificação. Num desses trabalhos, utilizou um processo de filtragem diferente, usando um filtro elíptico e uma classificação baseada num algoritmo de uma rede neuronal de Elman [11]. Os mesmos autores publicaram mais tarde, mais um artigo[12], onde se recorria à análise dos principais componentes (PCA) e compara a classificação *Simple Fuzzy ARTmap* (SFA) e *K-Nearest Neighbor* (kNN), constatando-se uma clara vantagem na utilização do primeiro algoritmo.

Realização experimental

3.1 Dados utilizados

Os dados utilizados desta tese, são provenientes de um outro projecto na área de neuropsicologia [17]. Esse projecto foi desenvolvido na área de neuropsicologia, nas universidades de Aveiro, Universidade Autónoma de Madrid e Universidade de York. No projecto de psicologia, o objectivo era investigar o papel da atenção no processamento intelectual de expressões de medo e nojo. Para desencadear este trabalho intelectual, foram usados estímulos visuais, e registados os respectivos PEVs. Apesar de o objectivo do projecto[17] ter uma finalidade bastante diferente do que é pretendido neste trabalho e o protocolo experimental ser idealizado para outros fins, pensa-se que os dados são susceptíveis de serem utilizados nesta tese. As imagens utilizadas como estímulos, não são todas iguais (foram utilizadas varias imagens diferentes para provocar os PEVs). A maneira como o cérebro processa as imagens é semelhante, por isso espera-se que os resultados não sejam afectados por este facto. Estas conclusões serão tomadas posteriormente, uma vez que foram tidas em consideração no processo de extracção de características. Será possível confirmar, se PEVs resultantes de diferentes estímulos, são ou não semelhantes o suficiente, para servirem de objecto de estudo na busca das características biométricas, que se procuram neste trabalho.

O procedimento experimental que originou os dados, que se pretende utilizar teve lugar na Universidade de Madrid. Dezasseis participantes do sexo feminino, com idades compreendidas entre os 20 e os 28 anos, todas destros, sem problemas neurológicos e com uma visão normal ou corrigida, foram sujeitas ao procedimento experimental de registo do sinal EEG.

3.1.1 Imagens utilizadas como estímulos

As imagens que serviram de estímulos visuais, foram em tons de cinzento de faces de dezasseis indivíduos (8 masculinos e 8 femininos). A estes indivíduos foi pedido que mostrassem uma expressão neutra, de medo e de nojo, resultando num conjunto de quarenta e oito estímulos faciais. Foram utilizados dois conjuntos de imagens para formar o conjunto dos dezasseis indivíduos. Foram utilizadas faces provenientes do set de Ekman e Friesen (Ekman & Friesen, 1976) e foi criado um set semelhante na Universidade Autónoma de Madrid (UAM set). No primeiro set encontram-se expressões de nove indivíduos (cinco masculinos e quatro femininos), e no UAM set estão presentes as imagens dos sete restantes (três masculinos e quatro femininos). A razão para terem sido utilizadas todas estas expressões faciais, de todos estes dezasseis indivíduos, prende-se com o facto de não se pretender que exista, redução no numero de repetições de estímulos faciais.

Para além das imagens faciais, foram recolhidas dezasseis figuras de diferentes casas, de diversas fontes da internet.



Fig3.1 Exemplo de estímulos resultantes da sobreposição entre as casas e as faces, com variação das expressões faciais: (A) Neutro; (B) Medo; (C) Nojo

O conjunto de imagens de faces e casas, foi dividido em dois grupos, A e B. Posteriormente foram sobrepostas faces com as casas, resultando num total de 384 estímulos, divididos em dois grupos de 192 imagens (8 casas por 8 indivíduos por 3 faces cada).

3.1.2 Aquisição de Dados

A experiência teve lugar numa sala ligeiramente iluminada, onde os participantes se encontravam sentados em frente a um ecrã, aproximadamente a 80cm de distância. Os estímulos foram apresentados com o software *Mind Tracer* (Neuronic S.A.), ligado a um computador equipado com software *Track Walker* (Neuronic S.A.), para o registo do EEG. As imagens (semelhantes às da Fig3.1), tinham 9.5cm de comprimento e 14cm de altura.

A experiência era constituída por quatro blocos experimentais. Em dois desses blocos, pedia-se aos participantes que se focassem nas caras, de modo a distinguirem se o sujeito da imagem que lhes estava a ser apresentada, difere do da imagem anteriormente observada. Nos dois blocos restantes, era pedido que se concentrassem nas casas e à semelhança dos dois blocos anteriores, a tarefa proposta era que os sujeitos comparassem a imagem com a anterior indicando se a casa era a mesma. A resposta do participante, era registada após um clique no rato do computador (situado sobre a mão direita), no botão direito ou esquerdo.

O motivo pelo qual existem dois blocos experimentais para faces e casas, reside no facto de, para cada (faces e casas), foi feita uma divisão entre expressões de medo e nojo, deste modo, em um dos blocos temos expressões de medo e normais, e num outro bloco temos nojo e normais. A justificação para esta divisão está relacionada com o objectivo central da experiência[17], onde se pretende estudar, comparativamente a resposta neuronal a um estímulo visual de uma expressão de medo e nojo, com uma face neutra.

Por cada bloco foram feitos 256 testes (8 indivíduos por 8 casas por duas expressões, apresentadas duas vezes). Os estímulos foram apresentados durante 300ms cada, e nos seguintes 500ms, depois de cada estímulo, era apresentado um *fixation cross*. Entre cada teste existia um tempo de latência de 2000ms. O sinal original de EEG é contínuo, isto é, 2000ms depois da apresentação de um determinado estímulo, era apresentado outro. Os dados fornecidos para a realização desta tese, não são de um sinal contínuo, uma vez que se encontra dividido em épocas de 1600ms, isto reflecte o facto de ter sido retirado porções do sinal que se consideram irrelevantes. As épocas encontra-se registadas consecutivamente.

3.1.3 Registo do sinal EEG

Para além do sinal EEG registado com 20 eléctrodo, com mais dois eléctrodos é registado o Electrooculograma (EOG). Esses dois canais, são os canais oculares, onde se encontra registada informação referente a movimentos involuntários da retina.

O sinal EEG encontra-se registado em vinte canais, segundo o modelo internacional 10-20 (fig. 2.1): o Fp1 e Fp2; F3 e F4; C3 e C4; P3 e P4; O1 e O2; F7 e F8; T3 e T4; P7 e P8; Fz, Cz, Pz e Oz. Estes sinais sofreram uma filtragem *on-line* durante o processo de registo. Para o efeito foi usado um filtro, com frequências entre os 0.05 e 30Hz (frequências de corte a 3dB).

Os eventos foram calculados *off-line*, constituídos por 400 pontos de digitalização por segundo, registados por intermédio de uma ADC de 12 bits (com uma frequência de amostragem de 250Hz). Os 400 pontos, correspondentes a uma amostra de 1600ms do sinal EEG. Destes 1600ms, 148ms (correspondentes a 37 amostras de sinal) são pré estímulo, seguindo-se de 1452ms pós estímulo.

Todos os eventos foram inspeccionados um a um e aqueles que apresentavam excessivos artefactos resultantes do sinal EOG, foram removidos. Posteriormente foi feita a média do sinal e por fim um acerto em termos de amplitude, realizado subtraindo a média do sinal correspondente ao pré-estímulo (os primeiros 148ms). É este sinal correspondente às médias globais que permite ver os potenciais evocados.

3.2 EEGLAB

O processamento dos dados EEG foi feito usando o programa MATLAB. A toolbox EEGLAB do MATLAB é uma ferramenta, que permite manipular com relativa facilidade o sinal EEG e mais importante, carrega os dados EEG para uma estrutura.

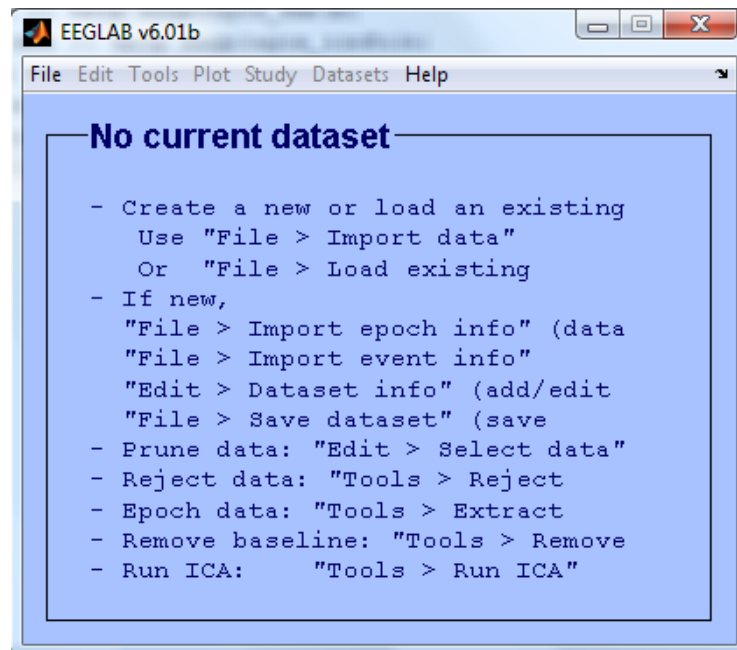


Fig3.2 Janela do EEGLAB

Para se fazer a leitura do sinal EEG recorreu-se ao EEGLAB, para isso basta abrir a toolbox e importar a *data* através dos comandos do programa.

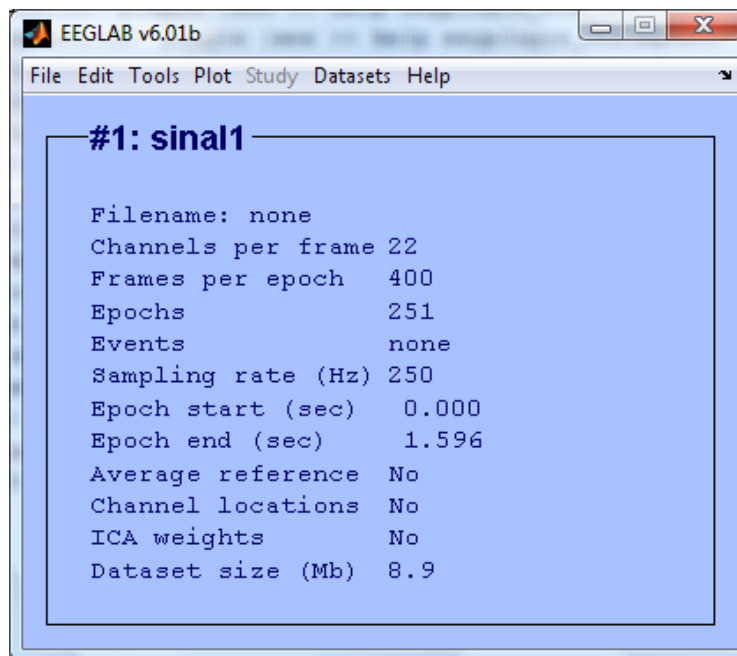


Fig3.3 EEGLAB carregado com os dados do primeiro sujeito

Depois de carregados os dados, definido a frequência de amostragem, número de amostras por época e de canais utilizados no registo do EEG, é constituída a estrutura de dados.

A estrutura definida apresenta vários parâmetros, entre eles o campo *EEG.data*, que corresponde à informação do EEG. Os dados apresentam as seguintes dimensões: *número de canais: número de amostras por evento: número de eventos*.

3.3 Identificação das características

Identificar significa encontrar, e é sob estas premissas que se pretende discriminar em cada evento de cada indivíduo, um conjunto de dados que possam ser atribuídos a determinada pessoa. Nesta fase pretende-se extrair do sinal EEG, um vector com valores específicos para cada indivíduo. Neste trabalho, esse vector de dados tem comprimento igual a 22, que corresponde ao número de canais usados para registar a resposta ao estímulo visual, juntamente com os canais EOG. Os canais EOG não são no entanto considerados para classificação, uma vez que nessa fase são removidos das bases de dados criadas para treino e teste do classificador.

Os valores que se pretendem extrair, são as energias na banda Gama de cada canal, no entanto existe um conjunto de alterações a que o sinal é sujeito, antes de determinarmos estas energias. As transformações que o sinal EEG sofre antes de serem determinadas as energias de todos os canais, encontram-se no diagrama de blocos da figura 3.4. Nos subcapítulos seguintes são explorados mais pormenorizadamente as técnicas referidas no diagrama de blocos.

Mais adiante é feita uma generalização de todo o processo de identificação de características, que conclui-se com a formação das bases de dados usadas na classificação. Também é referida uma outra abordagem ao processo de identificação, sem recurso à PCA.

3.3.1 Diagrama de blocos

O diagrama de blocos seguinte representa, de uma forma simplificada o processo de remoção das energias na banda Gama, do sinal original. O diagrama representa apenas a extracção das energias de um evento, isto é, as energias presentes no sinal EEG de apenas uma época, que corresponde à resposta de neurológica de um PEV.

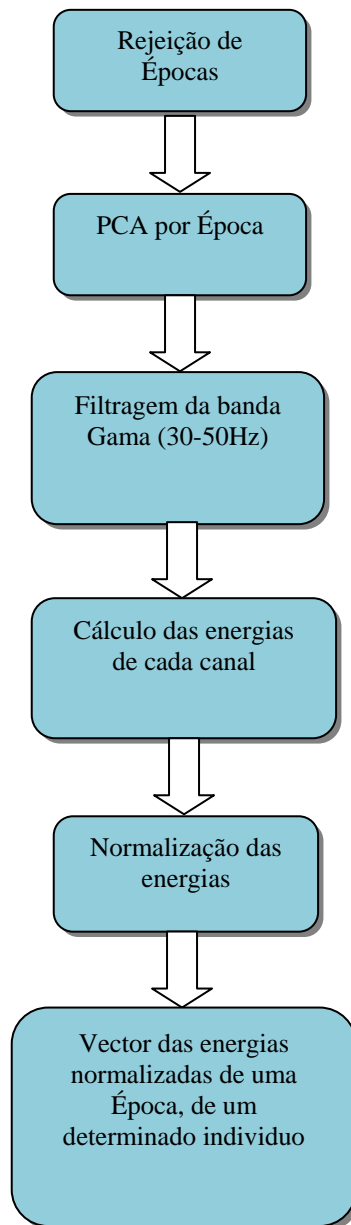


Fig3.4 Diagrama de blocos representativo do processo de extracção das características(vector de energias normalizadas) de um individuo

3.3.2 Rejeição de épocas com artefactos

Os dados utilizados neste trabalho correspondem à agregação das épocas, trata-se portanto de um conjunto de dados contínuo dentro de cada época, mas de um modo geral é descontínuo. Apenas se encontra no conjunto de dados, a informação correspondente ao registo do PEV, sem os dados inter-épocas, que não têm qualquer importância para este estudo.

No conjunto de épocas disponíveis, existem algumas que apresentam grandes variações de amplitude, que correspondem a artefactos. Estes artefactos necessitam de ser removidos, pois influenciam negativamente o resultado final. Estes eventos ao não serem removidos, induzem um erro no processo de classificação. Por essa razão, foi tomada a decisão de remover estas épocas. Estes factos são comprovados no capítulo 4, onde são apresentados os resultados experimentais.

As épocas seleccionadas para remoção, foram identificadas através de um processo demoroso de visualização de todo o *dataset*. O vector com as épocas rejeitadas de cada sujeito, para cada ficheiro de dados, foi determinado pelos responsáveis pela extracção do sinal EEG, de acordo com os seus critérios, de modo a aproveitar o maior número possível de épocas do *dataset*, removendo a informação corrompida com artefactos.

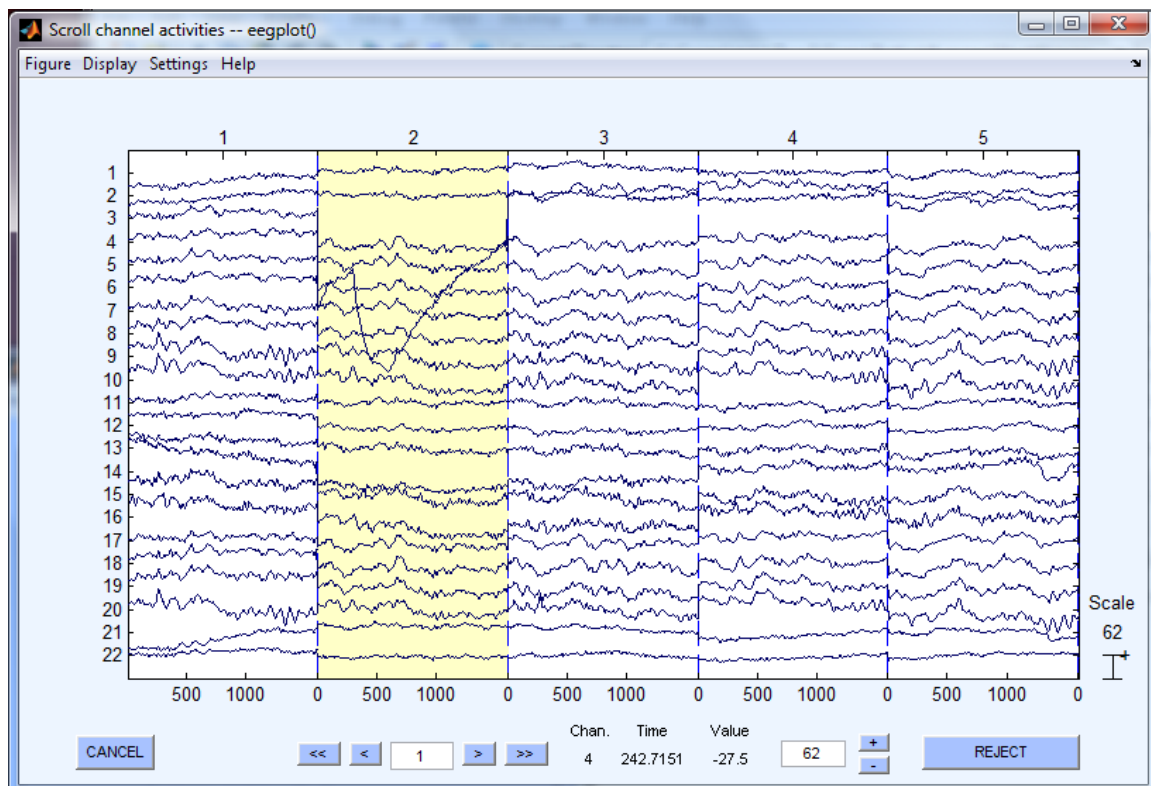


Fig3.5 Gráfico do sinal EEG do sujeito 1 das primeiras 5 épocas

Na figura 3.5, é possível visualizar as cinco primeiras épocas correspondente ao sujeito1 e são visível todos os vinte canais do sinal EEG e os dois EOG. É de realçar ainda o facto de a época 2 estar seleccionada, o que é visível através da diferente cor que apresenta. Esta época encontra-se seleccionada porque se trata de umas das épocas que têm de ser rejeitadas, isto é, removidas do conjunto de dados, que posteriormente servem de base para o segundo passo (PCA). Como se pode observar na figura 3.5, a época 2 apresenta bastante variação na amplitude em um dos seus canais, trata-se de um artefacto que vai influenciar o resultado das energias desse canal e consequentemente induzir erro no processo de classificação. Tal como foi referido anteriormente, esta época seleccionada, é uma das épocas a ser rejeitada.

Como é visível na figura 3.5 existe um botão na janela do interface do eeglab, que possibilita a rejeição manual de épocas, removendo os dados correspondentes a essas épocas. Este processo de rejeição, foi no entanto efectuado através de uma função criada para o efeito, de modo a simplificar e automatizar este processo.

Uma vez efectuado este processo de rejeição, verificou-se que para três sujeitos, o número de épocas rejeitadas devido à presença de artefactos é bastante grande. Nestes casos, o número de épocas rejeitadas é superior ao das épocas susceptíveis de aproveitamento. Foi devido a este facto que estas três pessoas foram removidas do universo de dados. Considerou-se que o número de épocas que se dispunha para classificação é demasiado inferior ao dos outros indivíduos, podendo isso influenciar o classificador devido ao reduzido número de objectos de treino.

3.3.3 PCA

A *Principal Components Analysis* (PCA), é um método matemático que transforma um número de variáveis possivelmente correlacionadas, em um menor número de variáveis não correlacionadas designadas de componentes principais. [19]. Este método permite, redução do número de dimensões sem perdas significativas de informação. Estas propriedades são bastante vantajosas, especialmente quando se lida com dados com múltiplas dimensões, tornando a PCA, num método estatístico muito útil.

Para melhor se compreender o funcionamento da PCA, primeiro é necessário conhecer alguns conceitos teóricos sob os quais assenta este método. Duas definições fundamentais são as de vectores próprios (*eigenvectors*) e valores próprios (*eigenvalues*).

Eigenvectors e Eigenvalues

Na álgebra linear, um *eigenvector* de um operador linear, é um vector não nulo que, quando operado por um operador, resulta em um escalar múltiplo de si mesmo. A este escalar chamamos *eigenvalue* associado ao *eigenvector*. A definição matemática diz que, C é um *eigenvector* de uma matriz A , se e só se, existir um número (real ou complexo) λ tal que:

$$AC = \lambda C \quad (3.1)$$

Se existir um λ tal que se verifique a equação 3.1, então esse valor é chamado de *eigenvalue* de A e C é o *eigenvector* associado a λ . É ainda de referir que C não pode ser nulo pois tem-se:

$$\mathbf{A} \cdot \mathbf{0} = \mathbf{0} = \lambda \cdot \mathbf{0} \quad (3.2)$$

para qualquer valor de λ , o que não é possível.

Os *eigenvectors*, são vectores com propriedades muito especiais:

- Um *eigenvector*, é um vector que quando é multiplicado por determinada matriz, não altera a sua direcção, apenas altera o seu tamanho;
- Os vectores podem ser dimensionados antes de serem multiplicados, uma vez que, o resultado será apenas um múltiplo do vector inicial, isto deve-se ao facto de apenas estarmos a alterar o comprimento do vector e não a sua direcção;
- Os *eigenvectors* de uma matriz simétrica ($\mathbf{A} = \mathbf{A}^T$) são ortogonais entre si, só para valores distintos de *eigenvalues*;
- O *eigenvector* principal de uma matriz é um *eigenvector* correspondente ao maior *eigenvalue* (para valores reais, maior valor absoluto).

Os *eigenvalues* também apresentam algumas propriedades:

- O valor absoluto de um determinante ($|\det \mathbf{A}|$) é o produto dos valores absolutos dos *eigenvalues* da matriz \mathbf{A} ;
- Se \mathbf{A} é uma matriz nxn triangular (superior ou inferior) ou uma matriz diagonal, os *eigenvalues* de \mathbf{A} são as entradas diagonais de \mathbf{A} ;
- A matriz \mathbf{A} e a sua transposta apresentam os mesmos *eigenvalues*;
- Os *eigenvalues* de uma matriz simétrica são todos reais;
- O *eigenvalue* mais pequeno de uma matriz \mathbf{A} , é o mesmo que o maior da matriz inversa \mathbf{A}^{-1} .

Depois de encontrados os *eigenvectors*, é costume redimensionarem-se os *eigenvectors* de modo a terem comprimento unitário. Como foi dito anteriormente, esta mudança de escala não influencia o facto de determinado vector ser ou não um *eigenvector* e deste modo fixa-se um valor standard para o comprimento.

Aplicação da PCA

A PCA serve como ferramenta para reduzir a dimensionalidade do *data_set* através de remoção dos componentes com menos conteúdo de informação. Com a PCA é possível remover-se algum ruído presente no sinal, funcionando como um filtro, resultando num sinal mais limpo.

Para determinar os *eigenvectors* (w) e os *eigenvalues* (D) associados foi criada uma função em Matlab denominada PCA, com o seguinte cabeçalho:

```
[W,D,Z1,m]=PCA(sinal, Evento);
```

Os argumentos de entrada são: a matriz de dados de determinado individuo (sinal) e o número da época que se pretende sujeitar a este processo.

Depois de seleccionado um evento, é formada a matriz de dados Z [22x400]. Este conjunto de dados é submetido a um processo de remoção da média por canal, m . A matriz resultante é $Z1$ e tem dimensão igual à da matriz Z .

O passo seguinte é determinar a matriz de co-variância C , de $Z1$, através da resolução da equação 3.4:

$$C = E(Z1 * Z1^T) \quad (3.3)$$

A matriz de co-variância C é dada pela média (operador E) do produto externo entre $Z1$ e $Z1$ ou, como se encontra representado na equação 3.3, pela média do produto interno entre $Z1$ e $Z1^T$. Esta operação é efectuada na prática através do produto interno entre a matriz $Z1$ e $Z1$ transposto, posteriormente dividindo, ponto a ponto, o resultado pelo número de pontos, isto é, as 400 amostras do evento. O produto interno, é por si só uma soma de produtos, portanto basta dividir-se o resultado pelo número de somas e chegamos ao resultado desejado, à co-variância desejada.

Seguidamente são calculados os *eigenvalues* e *eigenvectors*, para isso recorre-se a uma função *Matlab* intitulada de SVD (*Singular Value Decomposition*). Esta função recebe os dados presentes na matriz de co-variância (C) como parâmetros de entrada. A função SVD devolve uma matriz com os *eigenvectors* (W) e outra matriz diagonal (D), com dimensão igual à matriz de entrada. Os *eigenvalues* encontram-se organizados do maior para o mais pequeno, na diagonal principal da matriz D . Para além destes parâmetros a função SVD devolve um outro parâmetro o (O) que é uma matriz igual a W .

```
[reconst,R]=Eigenvalues(per, D, W, Zl,m)
```

Depois desta função PCA foi criada (em *Matlab*) uma função complementar, denominada de *eigenvalues*. O objectivo desta função, é proceder à remoção dos *eigenvectors* que possuem *eigenvalues* muito pequenos e reconstituir a matriz “inicial”, sem essa informação.

O critério usado para remoção de *eigenvectors*, foi o de remover os vectores em função da contribuição do correspondente *eigenvalue* no total dos *eigenvalues*. Um dos parâmetros de entrada, da função *Eigenvalues*, é a percentagem (per) de *eigenvalues* que se pretende, para a reconstrução do sinal. No caso desta tese foram considerados vários valores para este parâmetro, de modo a encontrar-se um equilíbrio entre os dados perdidos com a remoção destes vectores e a importância da informação que eles comportam para se atingir os objectivos propostos. Foram removidos os *eigenvectors* até se ficar com 95% dos *eigenvalues*. Na pratica foram somados os *eigenvalues* da diagonal da matriz *D*, por ordem decrescente (que é como está organizada) até se atingir 95% do valor total.

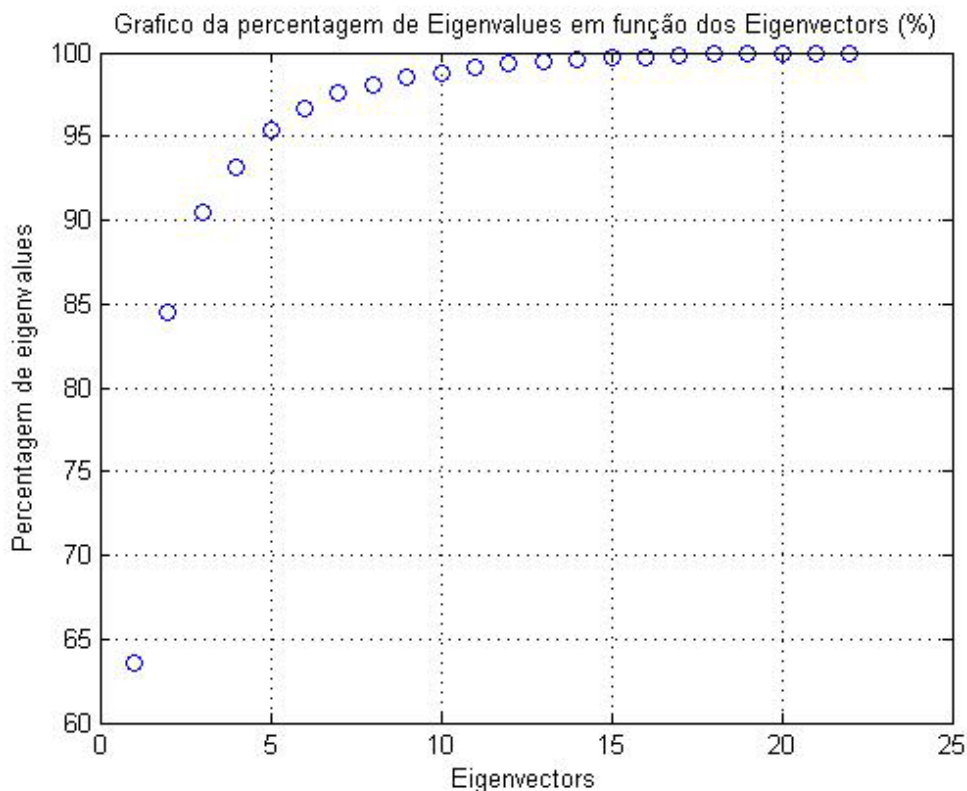


Fig3.6 Gráfico da percentagem de *eigenvalues* em função dos *eigenvectors*

O gráfico da fig3.6 representa a influência que os *eigenvectors* com maiores *eigenvalues* tem no conjunto dos dados. Como é visível no gráfico 3.6, para um determinado sujeito e evento, para atingir aproximadamente 95% do valor total de *eigenvalues*, são necessários 4 *eigenvalues*. Em seguida escolhe-se os *eigenvectors* correspondentes para projectar os dados. A escolha é baseada no valor dos *eigenvalues*.

O modelo é descrito pelos *eigenvectors*. Os restantes vectores não serão tidos em consideração na reconstrução da matriz.

Depois de determinados quais os *eigenvectors* que se pretende utilizar, reconstitui-se a matriz W , mas desta feita com apenas os 4 *eigenvectors* (esta matriz foi denominada de $W1$). A matriz $W1$, para este exemplo, apresenta dimensão [22x4]. De seguida, calculam-se os componentes principais, através da equação 3.4:

$$Y = W1' * Z1 \quad (3.4)$$

Por fim, para se obter a matriz reconstituída (que é variável de saída da função `eigenvalue: reconst`), multiplica-se a matriz $W1$ pela matriz Y da equação 3.4 e volta-se a “devolver” a média m , que tinha sido retirada na função PCA, daí o facto de ser um parâmetro de entrada da função.

Conclui-se assim a PCA com a obtenção da matriz reconstituída e são estes os dados, que vão ser processados pelos seguintes blocos de funções representados no diagrama de blocos da figura 3.4.

3.3.4 Filtragem da banda Gama (30-50Hz) e cálculo das energias dos canais

Nesta fase procede-se à filtragem da banda Gama (30-50Hz), que tal como foi referido anteriormente, é a banda de frequências que está relacionada com processos intelectuais, como a memória e resposta a estímulos visuais. Baseado neste facto e no facto de na bibliografia já existirem trabalhos onde se recorre a esta banda para estudar PEVs, optou-se por filtrar a banda Gama. Para esta filtragem, recorreu-se a um processo de filtragem, no domínio das frequências, que consiste na selecção das energias das frequências compreendidas dentro do intervalo da banda.

Para se efectuar esta filtragem, recorreu-se à função do *Matlab*, *pwelch* (*Power Spectral Density estimate via Welch's method*). Esta função devolve a densidade espectral de potência (DEP), da matriz de entrada, usando o método modificado de periodogramas de Welch.

No gráfico da fig3.7, é possível visualizar o espectro das energias do primeiro canal para o primeiro evento do sujeito1.

A grande vantagem de usar esta função, reside no facto de para além de ser possível a filtragem da banda Gama, também é possível determinar-se as energias de cada canal, para cada evento. A *pwelch* devolve a DEP por canal, que é o objectivo final desta fase de identificação, uma vez que são estas energias que são específicas ao indivíduo.

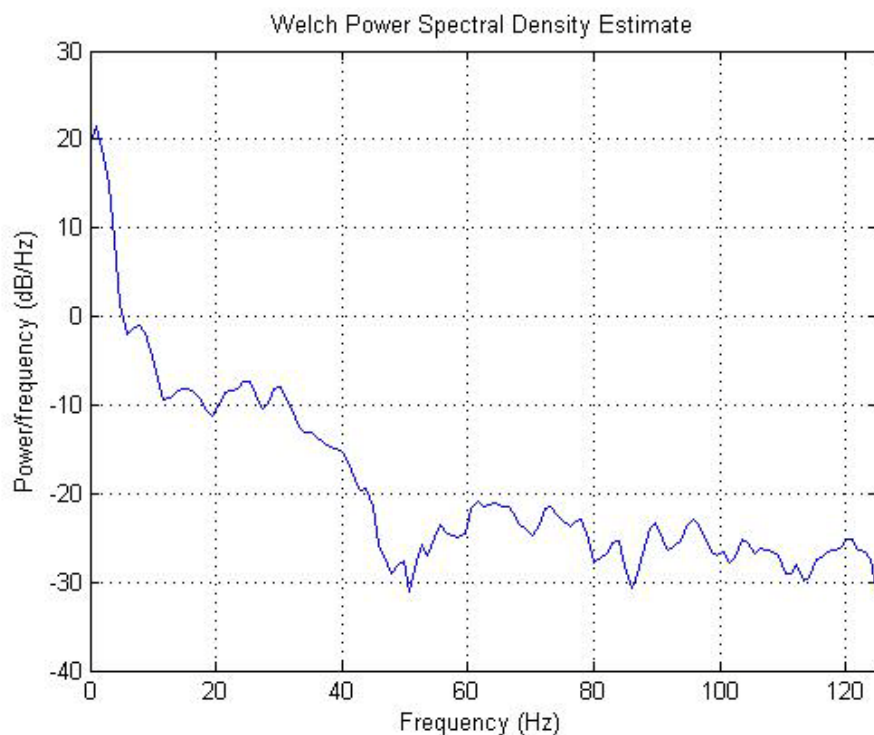


Fig3.7 Gráfico do resultado da função pwelch para um canal de um determinado evento

Aplicação da pwelch

Foi criada uma função chamada, pwelchfiltro. Esta função apresenta o seguinte cabeçalho:

```
[energ]=pwelchfiltro(A,B,reconst)
```

Como parâmetros de entrada da função, tem-se:

- A e B, que são os valores das frequências limite que definem o filtro. Neste filtro, todas as frequências serão seleccionadas deste A, a B inclusive. Mais especificamente, A e B tomam o valor de 30 e 50 respectivamente (banda Gama).
- reconst, é o outro parâmetro de entrada, que representa a matriz resultante da PCA, devolvida na função Eigenvalues, apresentada anteriormente. Apresenta dimensão [22x400], isto é, 22 canais por 400 amostras por evento.

Como parâmetro de saída, tem-se apenas um vector com 22 posições, que correspondem às energias dos 22 canais para determinado evento. Estas energias, depois de normalizadas, constituem as características singulares a cada individuo, que serão objecto de classificação.

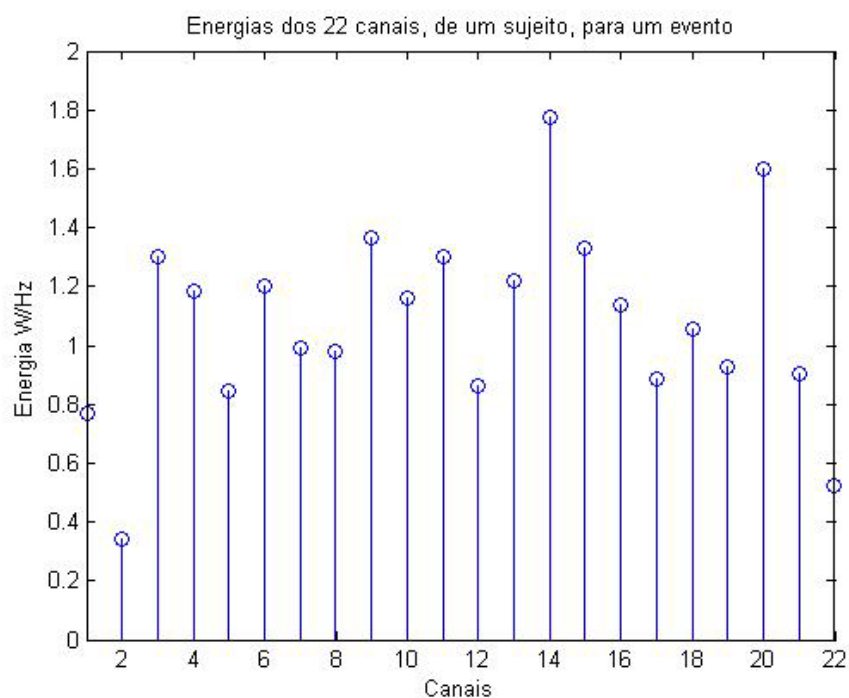


Fig.3.8 Gráfico das energias dos canais de um determinado indivíduo, para um evento

3.3.5 Normalização das energias

Normalizar as energias, significa escalar as mesmas, de modo a que a energia de cada canal seja uma parcela de um total da energia da banda Gama. Isto é conseguido dividindo a energia de cada canal pelo total das energias da banda, como mostra a figura 3.9:

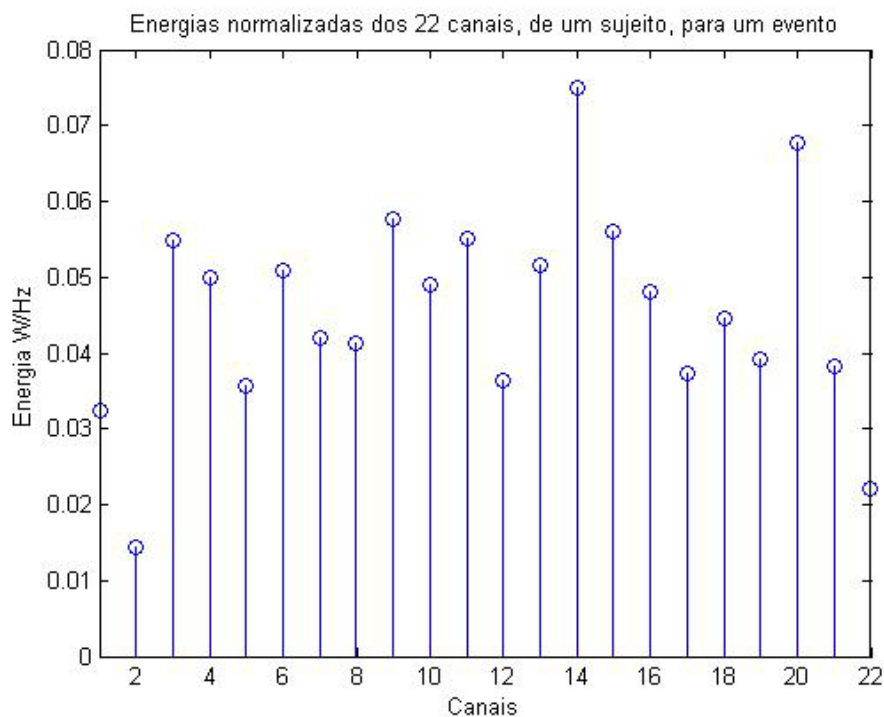


Fig.3.9 gráfico das energias normalizadas na banda Gama por canal

Este pequeno processo, foi incluído na função `pwelchfiltro`, dando origem à função `pwelchfiltro_norm`, que apresenta um cabeçalho semelhante ao da função `pwelchfiltro`.

Note-se que com a utilização da função `pwelch`, se simplificou o processo de identificação das características individuais, uma vez que foi possível implementar em apenas uma função, que comporta os blocos de: filtragem; cálculo de energias; e posteriormente, com uma pequena alteração, normalização.

3.3.6 Formação das bases de dados

Como foi referido anteriormente, o diagrama de blocos da figura 3.4 é apenas para identificar as energias para um determinado evento. Para se determinar estes valores de energia para todos os eventos de um determinado sujeito, foi criada uma função, cuja função é basicamente repetir ciclicamente o mesmo processo N vezes, em que N é o número de eventos previamente conhecidos de um ficheiro de dados de um sujeito. Este a rejeição de épocas é, obviamente mantida fora deste *loop*.

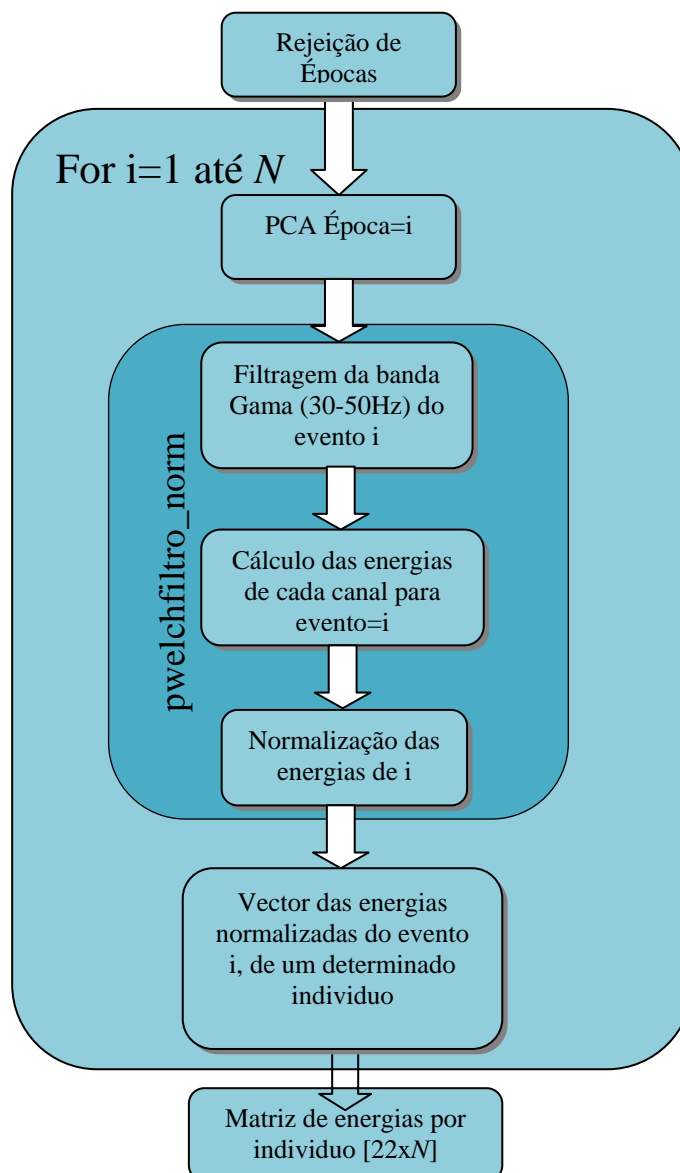


Fig.3.10 Diagrama de blocos do processo total de identificação das características

O diagrama da figura 3.10, representa o encadeamento de processos que levam à construção da matriz de dados com dimensão [canais x eventos]. São estes os dados que vão ser objecto de classificação. No diagrama de blocos 3.10 é possível perceber como é formada esta matriz. A matriz é vai sendo preenchida gradualmente até se terem esgotados todos os e eventos de um sujeito.

3.3.7 Formação das bases de dados, sem PCA

Para além do procedimento de extracção de características representado pelo diagrama da figura 3.10, foi seguida uma outra abordagem, bastante semelhante à anterior. A diferença relativamente ao passo anterior, reside no facto de neste caso não se recorrer à PCA. Desta feita, depois da rejeição de épocas com artefacto, passa-se imediatamente para a função `pwelchfiltro_norm`. Não existe mais nenhuma diferença relativamente ao método anterior, sendo os blocos representados no diagrama, iguais aos anteriores. Esta nova abordagem está representada no diagrama da figura 3.11.

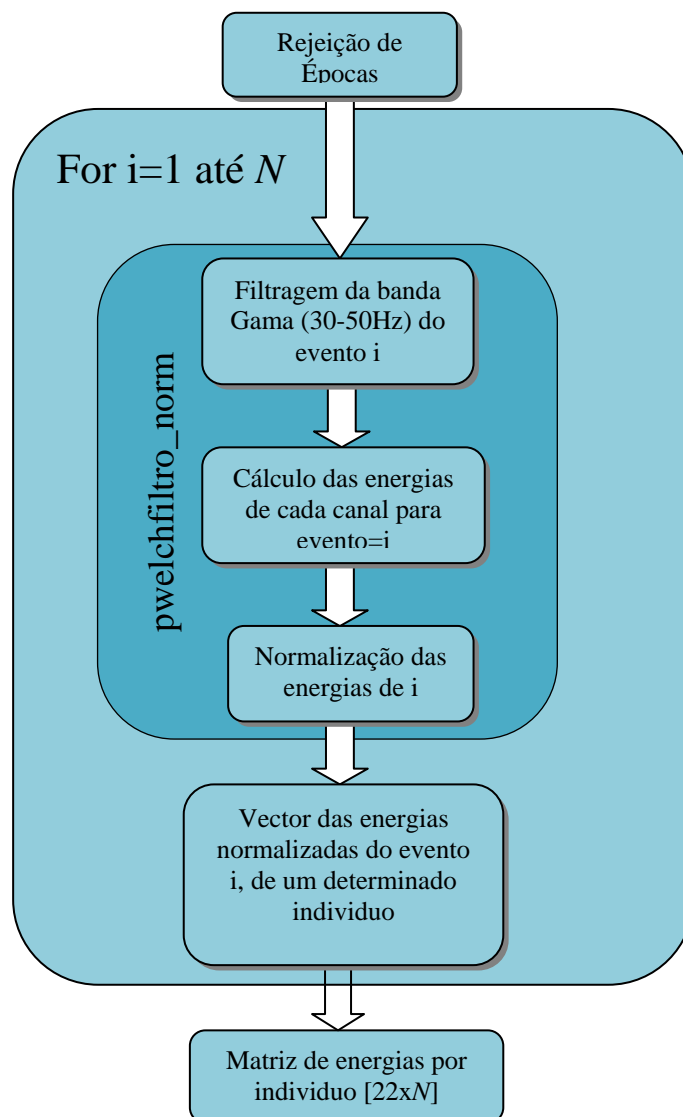


Fig.3.11 Diagrama de blocos relativo à identificação das características sem recurso à PCA

3.4 Classificação

Nesta parte do trabalho, foram criadas algumas funções, para que se consiga atribuir a um determinado individuo um vector de características correspondente (classificar). Na prática, o que se pretende é usar parte dos dados para treinar o classificador, de modo a que o classificador defina as suas margens para as diferentes classes. Os restantes dados disponíveis, são usados para testar o classificador. Estes dados utilizados, são dos vectores das características, determinados na parte de identificação. Existe um vector por cada época, para cada pessoa, variando o número de épocas por sujeito, devido ao número de épocas que foram rejeitadas anteriormente devido à presença de artefactos. É ainda de realçar relativamente aos dados utilizados para classificação, que os vectores das características encontram-se divididos, para cada pessoa, em dois ficheiros. Em um dos ficheiros, encontram-se o resultado do processo de identificação das características, para a experiência onde as pessoas tinham de se focar nas caras presentes nas imagens que serviram de estímulo. No outro ficheiro encontram-se as características das mesmas pessoas, quando lhes foi proposto se fixarem apenas nas casas das figuras. Tal como foi referido anteriormente, pensa-se que os resultados dos PEVs num e noutro caso seja semelhantes, mas no processo de classificação foi feita esta distinção, para que se possa tirar, conclusões acerca deste facto.

Este processo de classificação é baseado num classificador SVM, cujo método de funcionamento foi descrito no capítulo anterior. Foram usadas duas versões deste classificador. Um deles é o *Majority voting multi-class SVM* (MVSVM), em que, na fase de treino, o classificador efectua uma série de processos de classificação binária, para posteriormente decidir, por votação (a classe que foi mais vezes escolhida no processo anterior) a que individuo (ou classe) atribuir determinado evento, isto é, são definidas as “fronteiras” das classes.

A outra versão do classificador é a *One Against All decomposition* (OAASVM), onde na fase de treino, é efectuado um processo iterativo de classificação binária. Nesta fase, um conjunto de dados é confrontados com todos os restantes e depois, outro conjunto de dados é confrontado com que sobram. Este processo repete-se consoante o número de classes definidas.

Criaram-se diferentes funções para as duas versões do classificador. Para o classificador MVSVM foram usados métodos lineares e não lineares, tal como no OAASVM. Foram criadas funções distintas para aplicar estes classificadores, sendo no entanto bastante semelhantes. Em todas as funções eram efectuados os seguintes processos:

- Leitura das características, de um número definido de pessoas, no universo das treze disponíveis (não foram usadas as três pessoas com reduzido número de vectores de características, tal como descrito acima, na fase de rejeição);
- Remoção das energias correspondentes aos dois canais oculares, que não tem qualquer relevância para este estudo (uma vez que apenas servem para ajudar a identificar artefactos);
- Formação das estrutura de treino e teste, de acordo com as especificações do classificador e de acordo com o que é pretendido. Por vezes são usadas metade das características do ficheiro das casas e metade das caras,

para treino. Num outro caso são utilizadas unicamente caras para treino e casas para teste. Para estes dois casos existem duas funções distintas que apenas diferem neste método de constituir os parâmetros de entrada do classificador;

- Classificação de características relativas a um número de eventos (previamente definido), usando classificação linear ou não linear consoante a função.

```
[erro_inicial,ypred]=func_class_svm(caract,n_pessoas,eve_var)
```

As estruturas de dados (treino e teste) que servem de parâmetro de entrada ao classificador SVM, necessitam estar organizadas com a seguinte dimensão: [Eventos,20]. O número de eventos é variável, uma vez que é definido como parâmetro de entrada (eve_var) da função. O número 20 significa o número de canais usados para classificação, uma vez os oculares não foram considerados. É possível usar o classificador para um número de pessoas inferior a 13 e superior a 2, através da manipulação do parâmetro n_pessoas. Para além da formação das estruturas com as características, é necessário criar-se um vector onde se indica a classe a que determinado evento, da matriz de teste, pertence. Esta informação é necessária para que seja possível avaliar a eficiência do classificar. Esta avaliação é conseguida comparando o resultado da sua decisão, relativa a determinado evento, com a pessoa a que esse evento realmente pertence.

A função func_class_svm, devolve como parâmetros de saída duas variáveis. Uma dessas variáveis é erro_inicial, que é o valor da probabilidade de o classificador se enganar, isto é, é o número de eventos que o classificador se enganou a classificar, sobre o total de eventos sujeitos ao processo. Cada vez que o classificador classifica determinado evento como pertencente a outra pessoa, está a cometer um erro. O classificador sabe quando efectuou um erro, uma vez que é na construção das matrizes de treino e teste, são formados também dois vectores com a informação de que pessoa (classe), determinado evento pertence. Cruzando este vector informativo com o da previsão do classificador, é possível detectar os erros. A outra variável que a função devolve é a ypred. Esta variável é o resultado da previsão do classificador, trata-se de um vector com o comprimento igual ao número de eventos usados para teste. Este vector contém as decisões do classificador e esta informação é também importante para a fase seguinte desta tese, o pós-processamento.

Para melhor se compreender como funciona este método de classificação, usou-se o exemplo representado na figura 3.12. Foi efectuada a classificação para cinco pessoas e usou-se dez épocas por pessoa. Os dados foram constituídos de modo a que as primeiras dez épocas correspondam à primeira pessoa e sucessivamente, até às últimas dez épocas que correspondem ao individuo cinco. O classificador usado foi o MVSVM não linear.

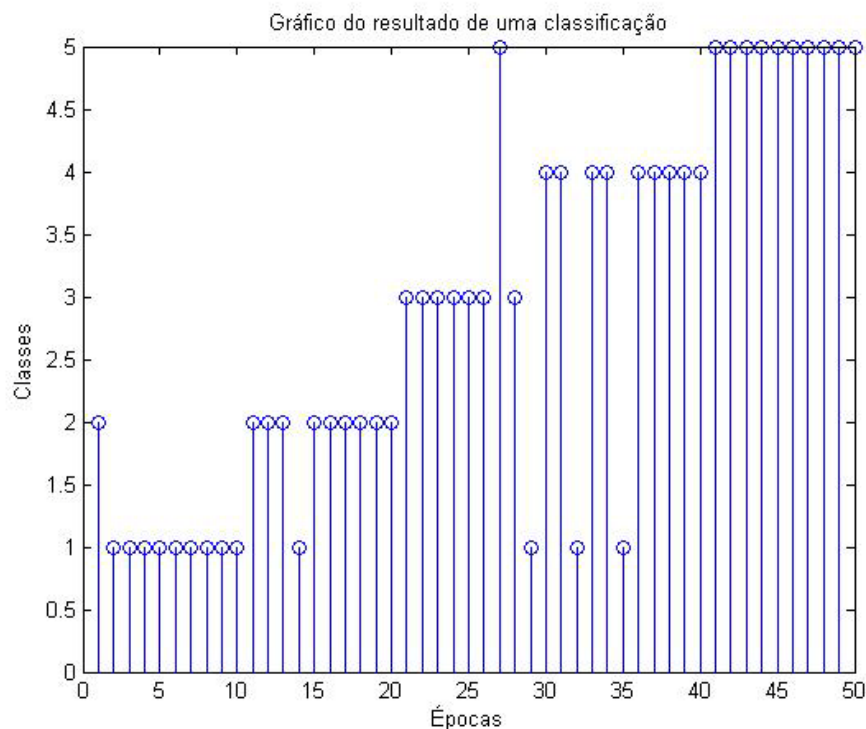


Fig.3.12 Gráfico de uma classificação com 5classes

O número de pessoas corresponde ao número de classe. Nesta classificação o erro foi de 14%. São visíveis os eventos onde o classificador se enganou, exemplo disso é o primeiro evento. No primeiro evento, o classificador previu que pertencia à pessoa dois, no entanto, como foi referido, todos os dez primeiros eventos pertencem ao primeiro sujeito. No caso da pessoa cinco, o classificador não cometeu qualquer erro, como se pode observar na figura 3.12.

O método usado para construir as matrizes de treino e teste, foi escolhido de modo a que, para determinado individuo, todos os eventos que se pretendem classificar se encontrem consecutivos. Foi escolhido esta metodologia, pois assim é possível efectuar uma correcção ao vector de previsão (ypred), do classificador, de modo a reduzir ainda mais o erro final. Este método é abordado no subcapítulo seguinte, no pós-processamento.

3.5 Pós-Processamento

Este sistema de pós-processamento, é uma maneira de tentar corrigir eventuais erros de classificação. A classificação é feita através de um continuo processo de classificação de eventos para a mesma pessoa. O pós-processamento visa corrigir os erros de decisão de alguns eventos, dentro da janela de eventos de teste previamente definida. Isto é conseguido através de uma análise das decisões anteriores e, em função destas altera-se ou não, a decisão do classificador. Este método é dividido em várias iterações, começando com uma análise de apenas três elementos do vector de decisão, aumentando progressivamente até uma janela de sete elementos.

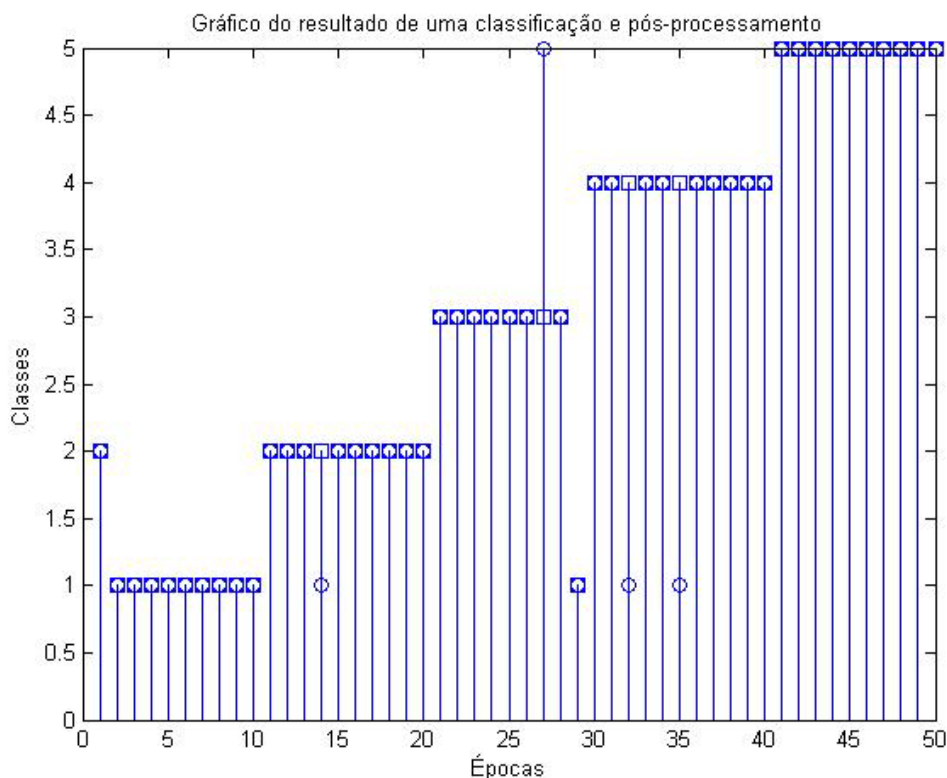


Fig.3.13 Gráfico de uma classificação e da correcção da mesma usando a primeira iteração do pós-processamento

O elemento que sofre alteração, no caso da primeira iteração, é o elemento central. É feita uma comparação entre os elementos que se encontram imediatamente antes e depois (janela de três elementos). Caso o elemento central seja diferente dos outros dois e se estes foram iguais entre si, procede-se à alteração.

No gráfico da figura 3.13 encontra-se o mesmo exemplo da figura 3.12, mas desta feita sobrepõe-se o resultado da primeira iteração do pós-processamento. No gráfico da fig3.13 encontram-se identificados com um círculo o resultado da classificação, e com um quadrado as decisões do pós-processamento. Pode-se observar que para o primeiro evento, não se verifica qualquer alteração, apesar do facto de o classificador se ter enganado. Isto é justificado pois a janela sendo de três elementos e como a alteração só é efectuada na época central dessa janela, este erro inicial e um eventual erro na última época nunca serão considerados. A janela de correcção é colocada inicialmente a partir da primeira época, com centro na segunda. Esta janela desloca-se sempre uma época para a direita, isto é, no passo seguinte o centro de decisão encontra-se na terceira época. Este processo desloca-se até atingir o final do vector, que acontece quando o centro da janela se encontra na penúltima posição. Sempre que é efectuada uma alteração, esta é tida em consideração quando a janela se desloca, pois as decisões tomadas são guardadas no vector que serve de alvo ao processo de pós-processamento. Outra limitação deste método é o caso onde existem dois erros consecutivos. Nesse caso não é efectuada qualquer alteração. Por esta razão criaram-se outras iterações, de modo a corrigir progressivamente a presença de dois, três, quatro, cinco e seis erros consecutivos.

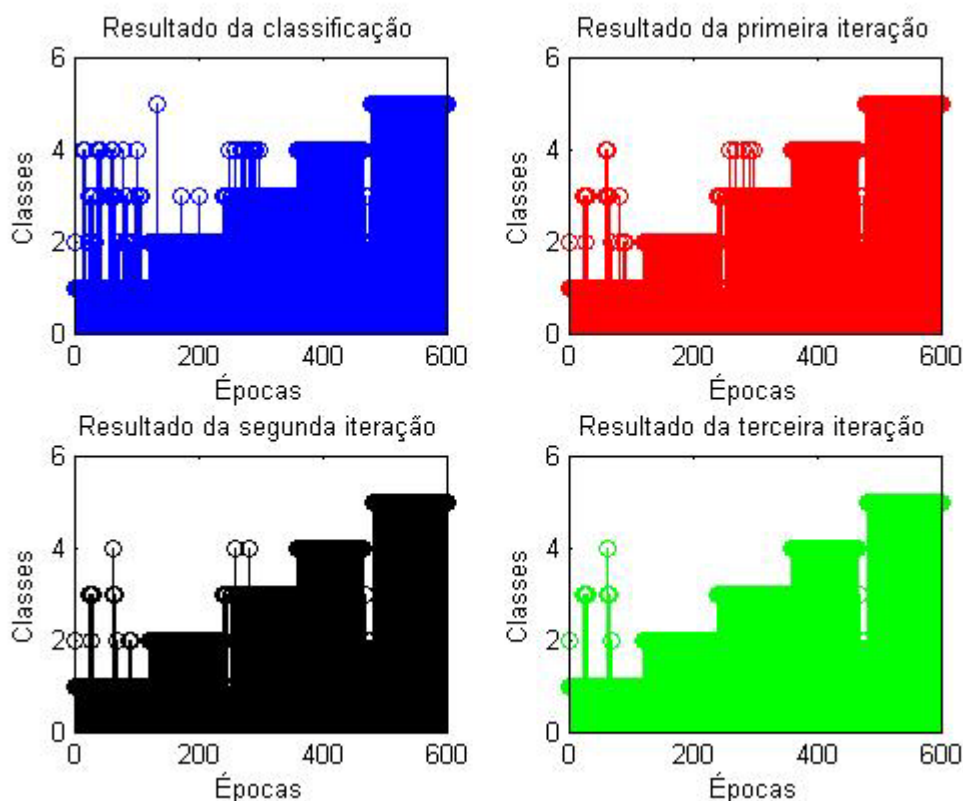


Fig.3.14 Gráficos do resultado de uma classificação e de três iterações do pós-processamento

No exemplo da figura 3.14, foram usadas também cinco pessoas e 120 eventos por pessoa para classificar. A figura pretende mostrar a diminuição do erro ao longo das três iterações. A melhoria gradual dos resultados é visível nos gráficos, pois para mais iterações há menos épocas mal classificadas. Em termos quantitativos, o erro antes do pós-processamento era de 20,67%, depois da primeira iteração de 11,67%, depois da segunda 7,67% e após a terceira de 6,17%.

O que difere nas seguintes iterações, relativamente à primeira é o tamanho da janela usada para decisão. Na segunda iteração a janela comporta quatro elementos. Neste caso são comparados os dois elementos centrais com os das margens da janela (primeiro e último). Caso os elementos dois e três sejam diferentes dos das margens e caso os das margens sejam iguais entre si, há alteração dos elementos centrais, ficando iguais aos marginais. Nas iterações seguintes o que aumenta é o número de elementos centrais. Aumenta deste um (primeira iteração), até cinco no último ciclo.

O uso deste método provou-se bastante eficaz na redução do erro total da classificação. Este procedimento cumpre os objectivos para os quais foi desenvolvido, que é a capacidade de “suavização” dos resultados da classificação, quando estes se encontram em sequencias consecutivas de épocas, por individuo.

Resultados Experimentais

Neste capítulo apresenta-se os resultados obtidos relativamente às abordagens que foram apresentadas nos capítulos anteriores. Nesta parte do trabalho será possível verificar se as opções que foram tomadas para atingir os objectivos propostos são correctas. Posteriormente, no capítulo seguinte poder-se-á tirar conclusões a propósito dos resultados quantitativos desta secção.

A apresentação dos resultados vai ser efectuada em dois subcapítulos, um para o caso onde se usou a PCA na identificação das características e outro sem a PCA. Dentro de cada subcapítulo são apresentados os resultados referentes à classificação dos diferentes classificadores.

4.1 Parâmetros do classificador SVM

Como foi referido anteriormente, aquando da explicação teórica dos princípios de funcionamento do classificador SVM, existem alguns parâmetros que necessitam de ser escolhidos. Esses parâmetros são: a função de Kernel; a constante de regulação C e σ que é um parâmetro da função de Kernel.

A função de Kernel utilizada é a *Radial Basis Function* (RBF), que foi enunciada no capítulo 2, na equação 2.18:

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / (2\sigma^2)} \quad (2.18)$$

Existem outras funções de Kernel possíveis, no entanto foi escolhida a RBF pois é a mais utilizada.

Quanto aos outros dois parâmetros (σ e C), estes são definidos através do uso da função pertencente à *toolbox* `stprtool`, denominada `evalsvm`. Esta função tem como parâmetros de entrada os dados de treino em uma estrutura onde se define o método de Kernel, uma função de classificação SVM (por exemplo a OAASVM); e os vectores com valores possíveis para σ e C . O classificador SVM é um classificador binário, no entanto existem métodos para usar o SVM para multi-classe, como é o caso da OAASVM (*One Against All SVM*) e da OAOSVM (*One Against Other SVM*). A função treina os dados com todos estes valores presentes nos vectores e decide sobre os melhores em função do valor de erro. Para o caso dos valores desta tese, foi determinado o valor de 0.1 para σ e de C igual a 10.

Para todos os classificadores SVM utilizados nesta tese, foram utilizados os parâmetros acima descritos. É de notar que no caso dos classificadores SVM lineares, não existe o parâmetro σ , pois este é proveniente da RBF, que só é utilizada no SVM não linear.

4.2 Resultados com PCA

A PCA, como foi apresentado no capítulo anterior, faz parte de um conjunto de operações que resultam na formação da base de dados, que é objecto de classificação. Estes resultados reflectem a classificação desta base de dados (energias de todos os indivíduos por todos eventos), quando usados os classificadores SVM.

4.2.1 Rejeição das épocas com artefactos

Pretende-se verificar, se existe alguma diferença nos resultados da classificação, se houver ou não remoção de artefactos. Para isso foi criada uma base de dados semelhantes à representada pelo diagrama de blocos da figura 3.13, mas desta feita sem o bloco de rejeição de épocas.

Para se proceder a esta comparação, recorreu-se à função `func_class_svm`. O classificador utilizado foi o MVSVM, desta feita na sua vertente não linear. São usadas as características onde se removeram os artefactos. Nestas condições, a classificação, para os treze indivíduos, usando as características resultantes da experiência onde se pede a atenção à alteração das caras, para treino e das casas para teste, obteve-se um erro de 43.93%. Verificou-se ainda que o erro depois de cinco iterações do pós processamento é de 23%.

No caso em que foi usada a base de dados, em que não foram removidos os artefactos, o erro de classificação é de 47.47% e o erro após a quinta iteração do pós-processamento é de 27.11%.

Para melhor se compreender a diferença entre os dois casos (com e sem artefactos), expôs-se no gráfico da figura 4.1 os resultados de um e outro caso, com e sem pós-processamento.

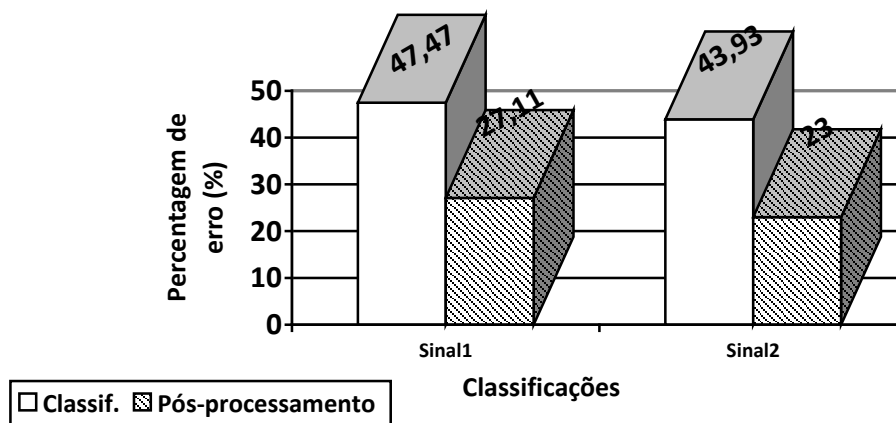


Figura 4.1 Gráfico dos resultados de classificação e pós-processamento, para os sinais com (sinal1) e sem artefactos (sinal2)

Através da observação dos resultados, constata-se que existe uma ligeira vantagem na remoção de artefactos. Os resultados mostram que existe uma discrepância de aproximadamente 4% entre a remoção e não remoção dos artefactos.

4.2.2 Classificador MVSVM não linear

Alguns dos resultados sobre o classificador MVSVM não linear, já foram apresentados na secção anterior. Os resultados expostos correspondem ao caso em que foram usados todos os vectores de características, correspondentes a todas as épocas disponíveis, para as treze pessoas. Nesse caso usaram-se todos os dados referentes às experiências das caras, para a fase de treino e todos os relativos à das casas para testar a resposta do classificador.

Para termo de comparação, foi realizada uma classificação semelhante à anterior, onde também são usados todos os dados referentes a todas as épocas. A diferença relativamente à classificação anterior encontra-se no facto de terem sido usados metade dos vectores de características das caras, juntamente com metade das características das casas para treino. Os restantes dados foram usados para o teste. O objectivo desta experiência é treinar o classificador usando os diferentes PEVs, para posteriormente se poder concluir algo sobre os resultados. É possível a comparação dos valores em que o classificador apenas é treinado usando as caras e quando é treinado com caras e casas. A esta classificação designou-se de classificação alternada, uma vez que se alterna entre caras e casas para construir a matriz de treino e teste.

Na tabela 4.1 encontram-se os resultados onde foram usadas apenas os dados referentes às experiências com as caras para teste. Os dados da experiência das casas serviram para teste do classificador. A tabela 4.2 contem os dados dos erros resultantes da classificação usando os dados alternados, isto é, 50% dos dados das caras e 50% dos dados das casas para treino e os restantes para teste. Em ambas as tabelas é apresentado o resultado das cinco iterações do pós-processamento, de modo a ser possível a verificação da diminuição do erro ao longo das progressivas iterações.

Tabela 4.1 Resultados da classificação com MVSVM não linear, usando os dados das caras para treino e casas para teste

	Resultado erro classificação (%)	Erro 1ª Iteração (%)	Erro 2ª Iteração (%)	Erro 3ª Iteração (%)	Erro 4ª Iteração (%)	Erro 5ª Iteração (%)
Resultados	44,53	37,26	31,24	27,95	26,19	24,07

Tabela 4.2 Resultados da classificação com MVSVM não linear alternado

	Resultado erro classificação (%)	Erro 1ª Iteração (%)	Erro 2ª Iteração (%)	Erro 3ª Iteração (%)	Erro 4ª Iteração (%)	Erro 5ª Iteração (%)
Resultados	36,43	28,00	22,08	19,01	17,41	14,49

Juntando num único gráfico os resultados de ambos os métodos anteriores (uso das caras para treino ou caras e casas), é possível tirar melhores conclusões sobre os resultados obtidos.

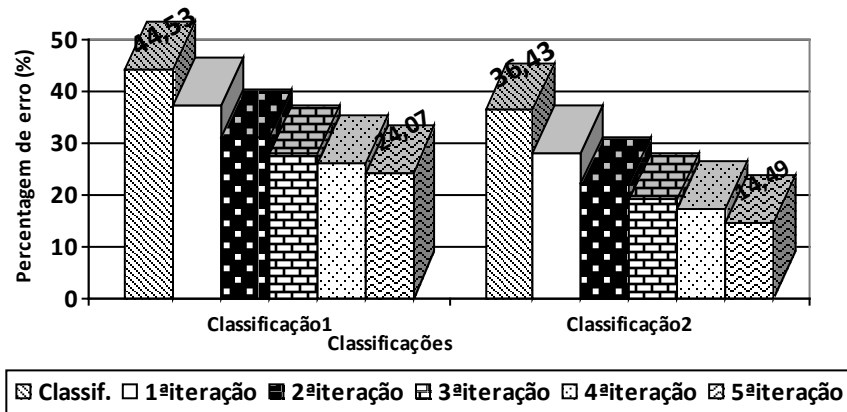


Fig4.2 Gráfico dos resultados da classificação do MVSVM não linear

No gráfico da figura 4.2, a Classificação1 representa a classificação usando as caras para treino e a Classificação2 o caso onde é feita a classificação alternada. Da análise do gráfico, e das tabelas 4.1 e 4.2, constata-se que o segundo caso apresenta menores valores de erro.

Para a classificação de um método de classificação individual baseada nestes sinais PVEs, importa conseguir reduzir o tempo necessário para o teste de cada indivíduo. Esta redução do tempo, durante o qual determinado indivíduo tem que ser sujeito ao registo dos sinais EEG é bastante importante para a viabilidade do sistema. No caso dos dados utilizadas, as épocas são de aproximadamente 1600ms, e torna-se importante saber a relação que existe entre o número de épocas testadas e o erro da classificação, de modo a minimizarmos o tempo de teste.

Nos casos anteriores, foram usados os dados de todas as épocas para treino e teste das treze pessoas. Nesta experiência, recorreu-se a uma função semelhante à que processa a classificação alternada, com a diferença de que processa apenas o número de épocas desejado. Escolheu-se esta função para se apresentar os resultados em função do número de épocas de teste, pois o caso onde o treino e teste são alternados, apresentou melhores resultados que o caso onde era feito o treino apenas com caras e o teste com casas.

A figura 4.3 representa o gráfico a partir do qual podemos tirar conclusões sobre o número mínimo de épocas necessários para uma classificação com um erro aceitável. O gráfico é o resultado de um conjunto de classificações em que o número de épocas de teste de cada classificação varia de 7 a 120. Na primeira vez que é efectuada o processo o número de épocas é 7 e na ultima vez 120. O valor inicial de 7 épocas para a primeira iteração, deve-se ao facto de a quinta iteração do pós-processamento necessitar de uma janela de 7 elementos. O valor final de 120 épocas foi escolhido pois é aproximadamente metade das épocas disponíveis para cada indivíduo e porque um número superior implicaria ainda mais elevados tempos de processamento computacional. Como é observável no gráfico da figura 4.3, existe um mínimo de percentagem de erro para aproximadamente 20 épocas de teste. A partir deste valor o erro aumenta ligeiramente consoante o número de épocas.

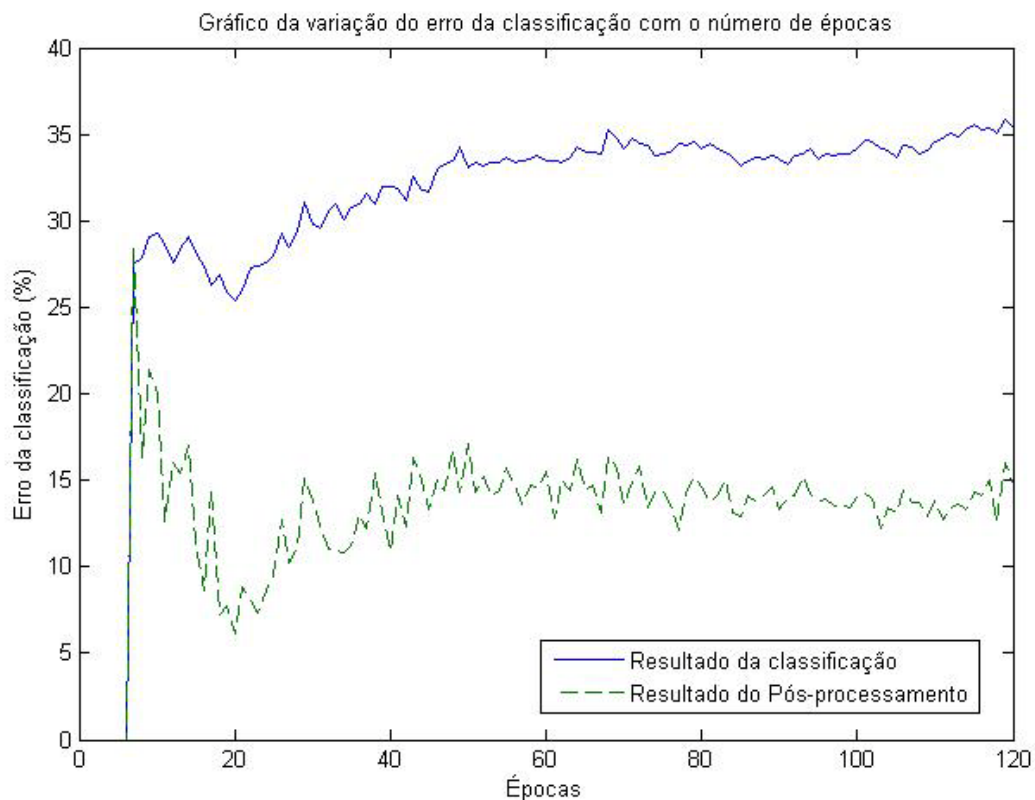


Fig4.3 Gráfico da variação do valor do erro da classificação alternada, desde 7 a 120 épocas.

4.2.3 Classificador MVSVM linear

Foram efectuados testes semelhantes aos do subcapítulo anterior, mas com o classificador MVSVM na sua vertente linear. Apesar do facto de muito dificilmente os dados serem linearmente separáveis, decidiu-se recorrer a este classificador para fins comparativos.

Nas tabelas 4.3 e 4.4 é possível consultar os valores utilizados para formar o gráfico da figura 4.4. Nestas tabelas pode-se comparar os resultados obtidos e analisá-los de acordo com a diferença nos valores do erro dos dois casos abordados.

Tabela 4.3 Resultados da classificação com MVSVM linear, usando os dados das caras para treino e casas para teste

	Resultado erro classificação (%)	Erro 1ª Iteração (%)	Erro 2ª Iteração (%)	Erro 3ª Iteração (%)	Erro 4ª Iteração (%)	Erro 5ª Iteração (%)
Resultados	65,94	63,10	60,01	59,71	59,79	59,61

Tabela 4.4 Resultados da classificação com MVSVM linear alternado

	Resultado erro classificação (%)	Erro 1ª Iteração (%)	Erro 2ª Iteração (%)	Erro 3ª Iteração (%)	Erro 4ª Iteração (%)	Erro 5ª Iteração (%)
Resultados	56,42	51,58	48,05	47,12	46,25	45,57

É usado o classificador MVSVM linear e é bem visível a discrepância entre os dados relativamente à mesma experiência, onde foi utilizado o mesmo classificador na forma não linear. Outro aspecto que não é bem visível na figura 4.4 é o facto de a quarta iteração do pós-processamento apresentar um erro superior à terceira iteração, para a classificação onde para a matriz de treino só são usados os dados da experiência das faces (Classificação1). Este facto já não se verifica para a classificação alternada (Classificação2).

A Classificação2 apresenta valores de erro inferiores comparativamente com os resultados obtidos quando o classificador é treinado com os dados das caras. Este facto também se verificou para o classificador MVSVM não linear, apesar de valores de erro para este classificador serem bastante superiores.

O gráfico da figura 4.10 mostra os resultados das classificações efectuadas com o classificador MVSVM linear. O objectivo é comparar os resultados dos dois tipos de classificações operadas, para se determinar o melhor método de construção das matrizes de treino e teste.

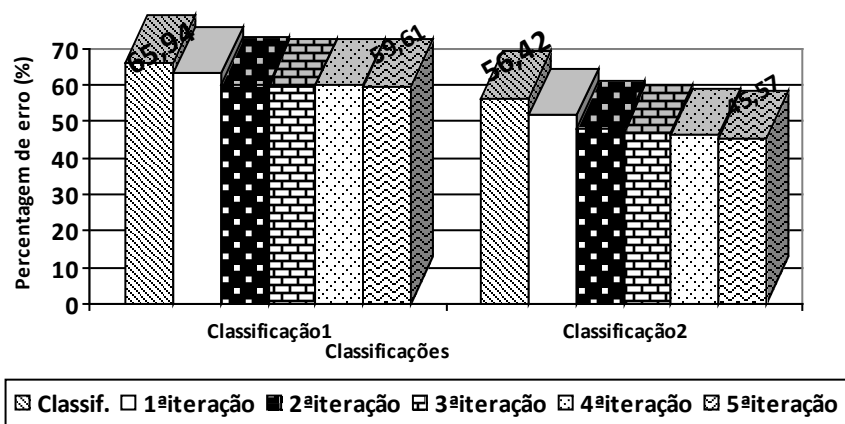


Fig4.4 Gráfico dos resultados da classificação do MVSVM linear

Pode-se constatar que os valores da classificação alternada são bastante inferiores aos da Classificação1. É possível observar também que a redução do erro devido aos vários processos iterativos do pós-processamento não se faz sentir tanto como no caso do classificador MVSVM não linear. Comparativamente aos resultados do MVSVM não linear, o linear apresenta erros bastante superiores. Tal como para o classificador MVSVM não linear, foi escolhido o caso onde se procedeu à construção das matrizes de treino e teste usando o método alternado, para a representação da evolução do erro consoante o número de épocas testadas.

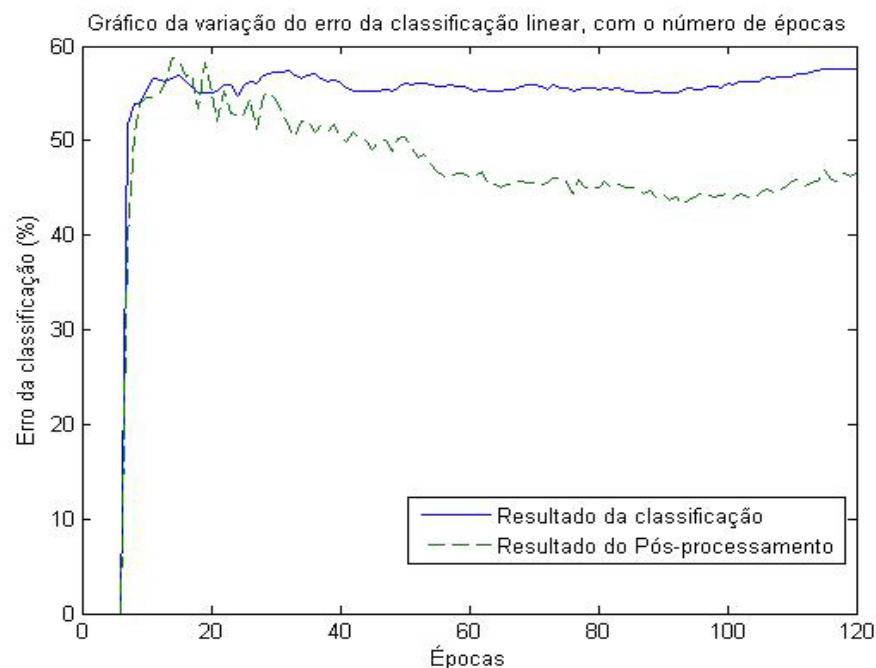


Fig.4.5 Gráfico da variação do erro da classificação alternada, desde 7 a 120 épocas.

O gráfico da figura 4.5 mostra que para a classificação, o valor do erro mantém-se aproximadamente constante, com um ligeiro aumento à medida que se testa o classificador com mais eventos. Quanto ao gráfico da quinta iteração do pós-processamento, é visível momentos em que toma valores de erro superiores à

classificação correspondente. Este efeito dissipa-se com o aumento do número de épocas, tendo uma diminuição cada vez maior do erro.

4.2.4 Classificador OAASVM linear

Para além do classificador MVSVM referido nos subcapítulos anteriores, também se usou o classificador OAASVM na sua forma linear e não linear. Os resultados seguintes correspondem à sua utilização à sua vertente linear.

Tomando a metodologia adoptada anteriormente, primeiro classificou-se todas as características disponíveis construindo a matriz de treino. Primeiro a matriz de treino foi constituída com os dados das caras em e posteriormente com caras e casas.

As tabelas 4.5 e 4.6 contêm os resultados presentes no gráfico de barras da figura 4.6.

Tabela 4.5 Resultados da classificação com OAASVM linear, usando os dados das caras para treino e casas para teste

	Resultado erro classificação (%)	Erro 1ª Iteração (%)	Erro 2ª Iteração (%)	Erro 3ª Iteração (%)	Erro 4ª Iteração (%)	Erro 5ª Iteração (%)
Média	58,65	54,24	50,64	49,88	49,34	48,60

Tabela 4.6 Resultados da classificação com OAASVM linear alternado

	Resultado erro classificação (%)	Erro 1ª Iteração (%)	Erro 2ª Iteração (%)	Erro 3ª Iteração (%)	Erro 4ª Iteração (%)	Erro 5ª Iteração (%)
Resultados	59,79	56,55	54,42	53,42	52,36	51,24

A figura 4.6 permite comparar-se os dois métodos anteriores, através dos seus resultados médios dos erros.

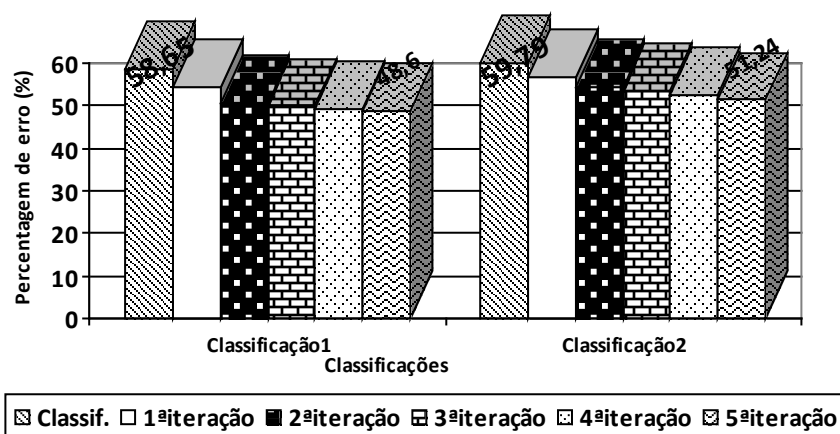


Fig.4.6 Gráfico dos resultados da classificação do OAASVM linear

A Classificação1 trata-se do caso onde o treino e teste usam dados de caras e casas e a Classificação2, o caso alternado. Os erros de classificação e pós-processamento, são ligeiramente superiores no caso da Classificação2. Este facto contraria o que tinha vindo a acontecer com o classificador MVSVM, no entanto a diferença de 1.14% para o resultado da classificação e 2.64% para o pós-processamento, não é muito significativa.

Os resultados das classificações representadas na classificação alternada mostram que, o erro é ligeiramente superior ao presente na classificação usando o treino e teste separados (Classificação1). Quanto ao pós-processamento, já é mais evidente a melhoria dos resultados comparativamente ao caso anterior.

Os resultados apresentados para este classificador são ligeiramente superiores quando comparados com os resultados do classificador MVSVM linear, nas mesmas condições. Verifica-se ainda que o pós-processamento, à excepção da primeira iteração, tem pouco efeito na diminuição do erro da classificação. Este aspecto é recorrente, uma vez que para o classificador MVSVM linear, também se verificou algo semelhante.

Através do gráfico seguinte (figura 4.7) é possível visualizar o comportamento do valor do erro à medida que se aumentam as épocas de teste.

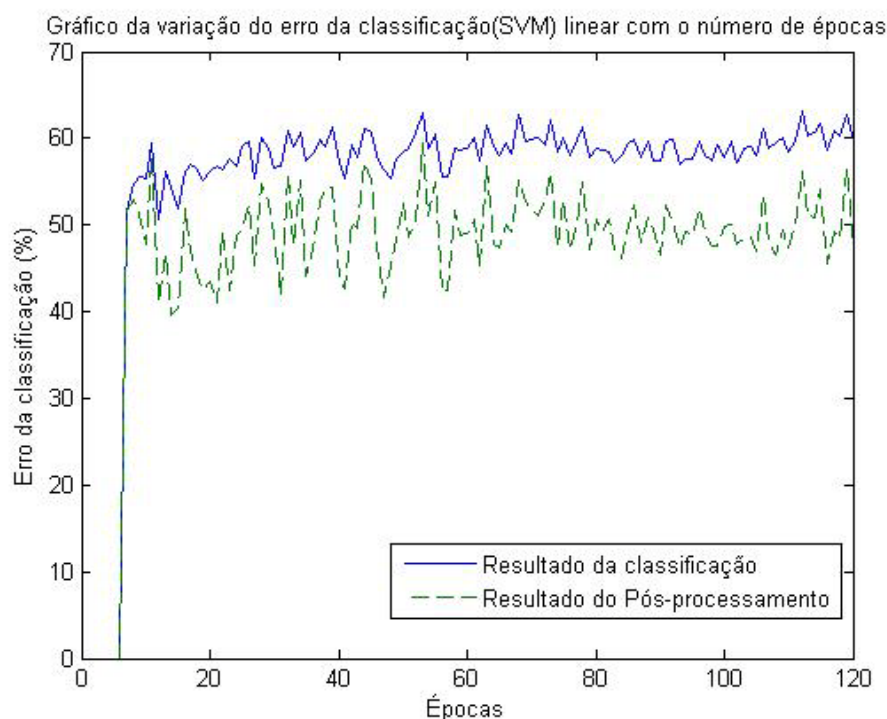


Fig.4.7 Gráfico da variação do erro da classificação OAASVM linear, desde 7 a 120 épocas.

No gráfico da figura 4.7 verifica-se uma grande oscilação do valor do erro, mais acentuada no gráfico do pós-processamento. Esta oscilação tende a diminuir à medida que o número de épocas aumenta, devido à diminuição na oscilação no erro do resultado da classificação. De um modo geral o erro mantém-se constante e bastante elevado.

4.2.5 Classificador OAASVM não linear

Apresentam-se de seguida os resultados referentes ao classificador OAASVM na sua forma não linear.

No gráfico 4.8 apresentam-se os resultados das classificações com o classificador treinado com apenas dados pertencentes às experiências com as caras (Classificação1) e treinado com o método alternado (Classificação2).

De seguida apresentam-se as tabelas 4.7 e 4.8, que mostram os resultados obtidos com as duas experiências anteriores:.

Tabela 4.7 Resultados da classificação com OAASVM não linear, usando os dados das caras para treino e casas para teste

	Resultado erro classificação (%)	Erro 1ª Iteração (%)	Erro 2ª Iteração (%)	Erro 3ª Iteração (%)	Erro 4ª Iteração (%)	Erro 5ª Iteração (%)
Resultados	43,78	36,76	31,12	28,03	24,93	23,33

Tabela 4.8 Resultados da classificação com OAASVM não linear alternado

	Resultado erro classificação (%)	Erro 1ª Iteração (%)	Erro 2ª Iteração (%)	Erro 3ª Iteração (%)	Erro 4ª Iteração (%)	Erro 5ª Iteração (%)
Resultados	35,99	27,60	21,24	18,67	16,44	15,17

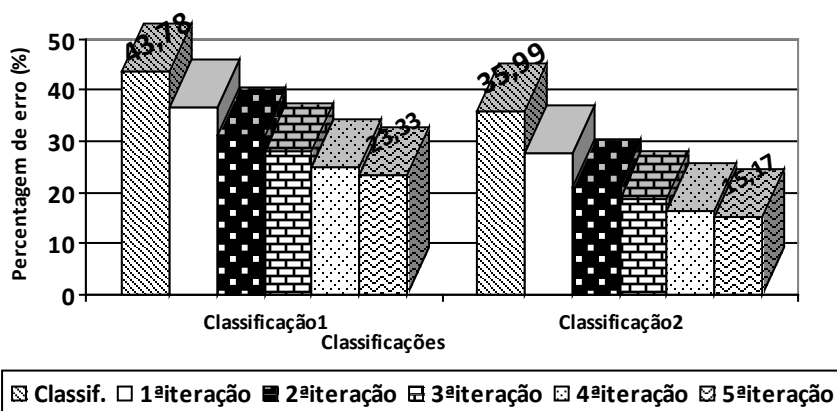


Fig4.8 Gráfico dos resultados da classificação do OAASVM não linear

Os resultados observados no gráfico da figura 4.8, são bastante melhores quando comparados com os do mesmo classificador linear. É de realçar também a significativa redução do erro com o pós-processamento, coisa que na vertente linear deste classificador, não é tão explícita. Estes valores de erro assemelham-se bastante aos resultados obtidos para o classificador MVSVM não linear.

Através da análise do gráfico da figura 4.8, constata-se que o valor do erro da classificação com treino alternado é inferior ao caso anterior (treino com dados das caras e teste com os dados das casas).

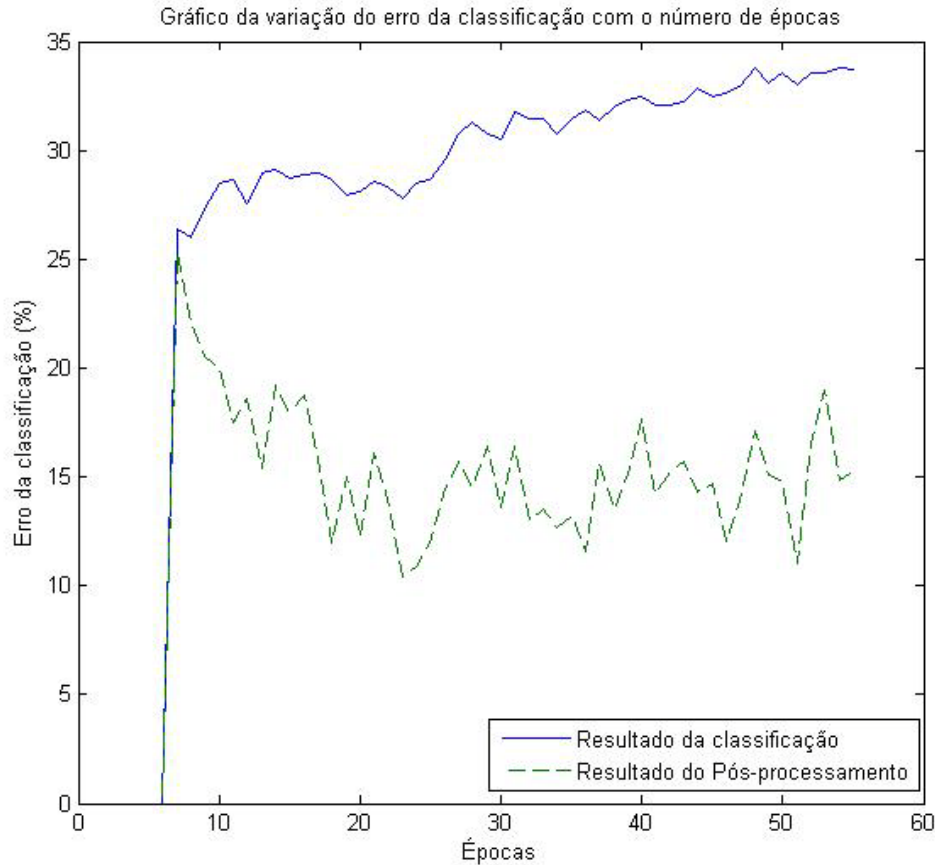


Fig.4.9 Gráfico da variação do erro da classificação OAASVM não linear, desde 7 a 55 épocas.

O número de épocas usadas para construir o gráfico da figura 4.9 varia de 7 a 55. Este valor 55 foi escolhido devido ao facto de este método ser muito exigente em termos computacionais e baseado nos resultados anteriores. Nos gráficos anteriores semelhantes, o valor mínimo de erro encontrava-se perto do intervalo entre as 20 e as 30 épocas. Quando isto não se sucede, o gráfico tende a crescer de uma forma contínua. Foi por estas razões que se decidiu diminuir o número de épocas máximo para 55.

No gráfico da figura 4.9 é visível um aumento do erro da classificação com o aumento do número de épocas. Este aumento é mais significativo do que o apresentado nos casos anteriores. Quanto ao pós-processamento, é visível um mínimo entre o intervalo das 20 e as 30 épocas e o aumento do erro não é tão significativo como no resultado da decisão do classificador.

4.3 Resultados sem PCA

Os resultados das classificações que vão ser apresentados, corresponde ao caso onde, para a identificação das características individuais, não foi utilizada a PCA. O método de exposição dos resultados é semelhante ao usado no subcapítulo anterior. Neste subcapítulo, são apresentados os resultados consoante o classificador usado.

4.3.1 Classificador MVSVM não linear

Primeiramente (Classificação1) são apresentados os resultados deste classificador treinado com os dados da experiência onde se pedia atenção à alteração das caras e testado com os dados de uma experiência semelhante mas para casas. De seguida apresentam-se os resultados do uso do mesmo classificador, mas treinado com dados alternados (Classificação2), provenientes dos dados das experiências das caras e casas. Nesta experiência, tal como na anterior, foram usados todos os dados, metade para treino e metade para teste, para as treze pessoas.

Comparativamente a todos os resultados apresentados anteriormente, os representados no gráfico da figura 4.9, são os que apresentam um valor de erro inferior. O erro da quinta iteração do pós-processamento chega a atingir valores que rondam os 15% para a Classificação1 e 5% para a Classificação2.

Os resultados da classificação2 são melhores que os obtidos para a Classificação1 (treino e teste com dados independentes). Existe uma variação nos resultados, mas de um modo geral as classificações obtiveram bons resultados. Quanto ao pós-processamento, revelou-se bastante eficaz, reduzindo bastante o erro. Estes são os melhores resultados obtidos, quando comparados com todas as abordagens anteriores.

Tabela 4.9 Resultados da classificação com MVSVM não linear, usando os dados das caras para treino e casas para teste

	Resultado erro classificação (%)	Erro 1ª iteração (%)	Erro 2ª iteração (%)	Erro 3ª iteração (%)	Erro 4ª iteração (%)	Erro 5ª iteração (%)
Resultados	26,42	20,31	17,42	16,87	15,97	14,95

Tabela 4.10 Resultados da classificação com MVSVM não linear alternado

	Resultado erro classificação (%)	Erro 1ª iteração (%)	Erro 2ª iteração (%)	Erro 3ª iteração (%)	Erro 4ª iteração (%)	Erro 5ª iteração (%)
Resultados	15,67	10,16	8,32	6,95	5,54	5,10

Neste caso, comparando os tipos de modelo de treino e teste do classificador MVSVM não linear, os resultados são mais expressivos quanto ao método que apresenta melhor resultados.

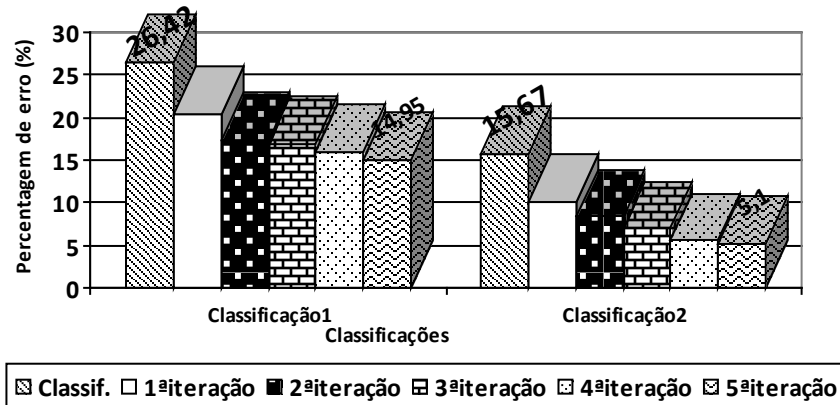


Fig4.10 Gráfico dos resultados da classificação do MVSVM não linear

Procedemos por fim à análise da variação do erro em função do número de eventos. Essa variação está representada no gráfico da figura 4.11.

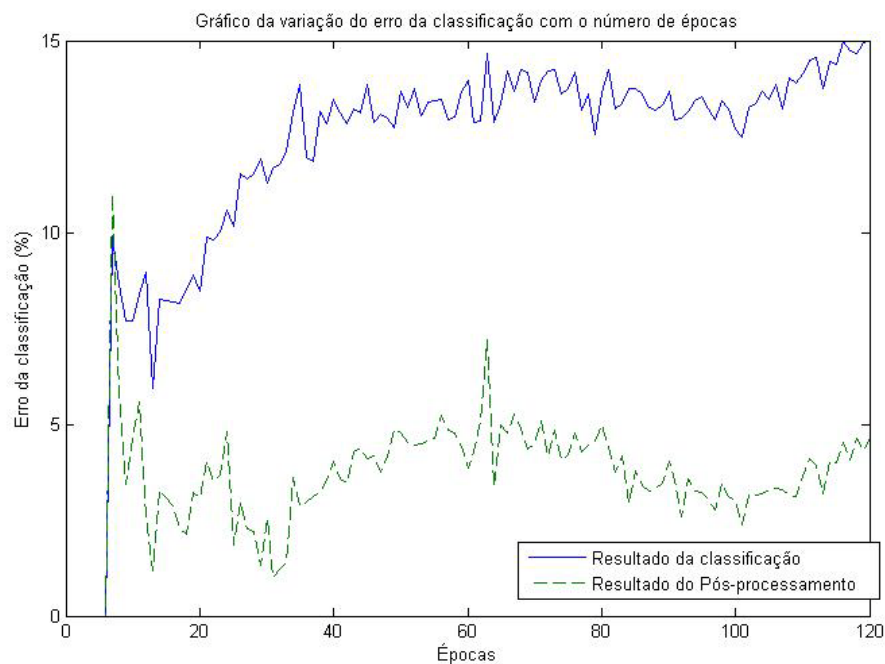


Fig.4.11 Gráfico da variação do erro da classificação MVSVM não linear, desde 7 a 120 épocas.

Neste gráfico é visível que os valores de erro, de um modo geral, aumentam com o aumento do número de épocas, apresentando um mínimo entre as 10 e as 20 épocas. Neste caso é visível o efeito do pós-processamento, apresentando uma grande diferença entre o resultado da classificação e os resultados da quinta e final iteração.

4.3.2 Classificador MVSVM linear

Os resultados que se seguem correspondem a classificações efectuadas com recurso ao classificador MVSVM linear, sem recurso a PCA para identificação dos vectores de características.

O gráfico da figura 4.12 apresenta resultados referentes a classificações com treino e teste com dados independentes (Classificação1), bem como os resultados do uso do mesmo classificador, mas com o treino e teste efectuados com dados alternados (Classificação2). Todos os dados foram usados, para todas as pessoas. Os resultados são também apresentados em duas tabelas (4.11 e 4.12) correspondentes a cada uma das experiências. Através do gráfico, pretende-se compreender melhor as diferenças entre os resultados dos dois métodos de construção das matrizes de treino e teste.

Finalmente exhibe-se no gráfico 4.13 a relação entre o valor de erro da classificação e o número de épocas utilizadas, de modo a se tentar perceber qual o número de épocas mínimo para implementação do sistema de identificação biométrico. Para este gráfico é usado o método de construção das matrizes de treino e teste, que apresenta melhores resultados. No caso em questão, tal como em quase todos os casos anteriores, trata-se do método alternado.

Os resultados visíveis no gráfico da figura 4.12 são bastante piores do que os apresentados com o mesmo classificador, mas na sua vertente não linear. No entanto os resultados são bem melhores quando comparados com os obtidos com o mesmo classificador linear no caso onde foi utilizada a PCA. Pode-se observar ainda que o pós-processamento não tem grande influência na redução do erro.

No gráfico da figura 4.12, é bem notória a disparidade dos valores de erro, relativamente aos resultados dos dois métodos em estudo. A Classificação2 apresenta resultados de pós-processamento aproximadamente 9% interiores aos da Classificação1. É ainda de destacar o facto de a terceira iteração do pós-processamento da Classificação2, apresentar valores de erro superiores aos apresentados na segunda iteração. Este erro introduzido é corrigido na quarta iteração, pois esta apresenta erros inferiores às iterações anteriores.

Tabela 4.11 Resultados da classificação com MVSVM linear, usando os dados das caras para treino e casas para teste

	Resultado erro classificação (%)	Erro 1ª Iteração (%)	Erro 2ª Iteração (%)	Erro 3ª Iteração (%)	Erro 4ª Iteração (%)	Erro 5ª Iteração (%)
Resultados	38,21	35,36	33,43	31,89	31,63	30,37

Tabela 4.12 Resultados da classificação com MVSVM linear alternado

	Resultado erro classificação (%)	Erro 1ª Iteração (%)	Erro 2ª Iteração (%)	Erro 3ª Iteração (%)	Erro 4ª Iteração (%)	Erro 5ª Iteração (%)
Média	29,98	24,88	23,19	23,55	22,77	21,54

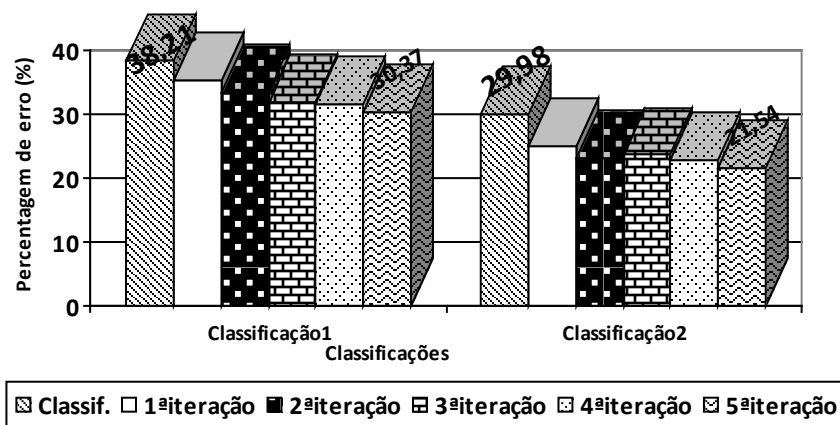


Fig4.12 Gráfico dos resultados da classificação do MVSVM linear

Na figura 4.11 podemos observar que, tal como foi referido, o gráfico correspondente à Classificação1 apresenta valores de erro superiores à Classificação2. A Classificação1 e 2 representam os métodos de construção da matriz de treino e teste não alternado e alternado respectivamente.

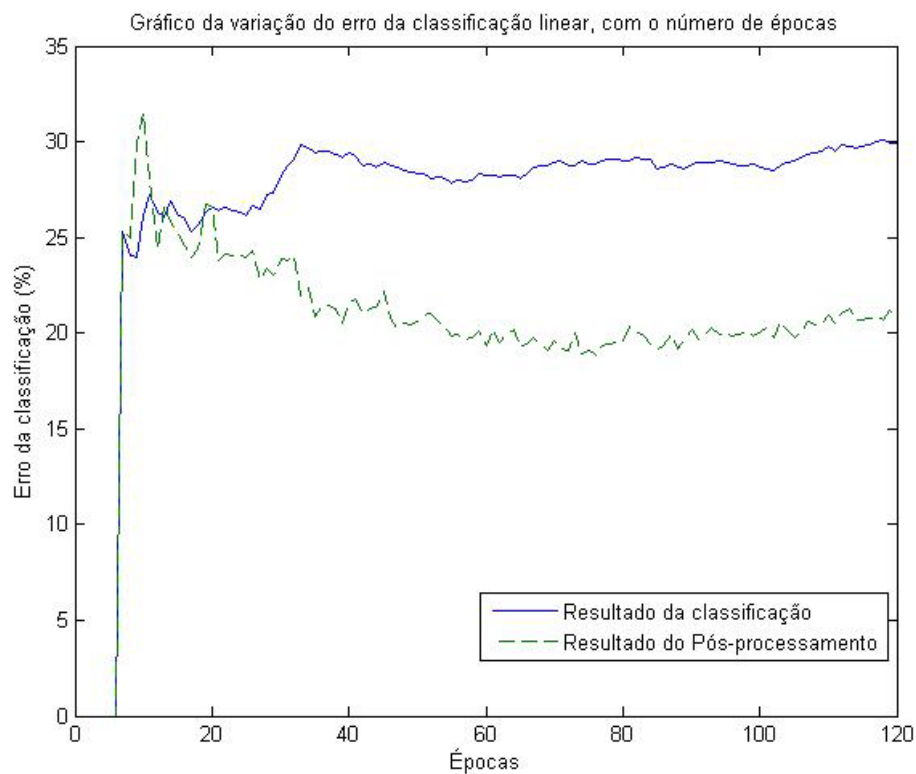


Fig.4.13 Gráfico da variação do erro da classificação MVSVM linear, desde 7 a 120 épocas.

O erro da classificação vai aumentando ligeiramente com o aumento do número de eventos usados para o teste do classificador. Quanto erro da quinta iteração do pós-processamento, este diminui progressivamente e converge para o valor médio apresentado no gráfico 4.12, que é aproximadamente 22%. Nas primeiras épocas apresenta um pico que ultrapassa o valor original da classificação, mas rapidamente se revela eficaz na sua função de redução do erro.

4.3.2 Classificador OAASVM linear

Neste subcapítulo apresentam-se os resultados correspondem a classificações operadas com o classificador OAASVM linear sem recurso a PCA para identificação dos vectores de características.

Na figura 4.14 apresentam-se os resultados da classificações com treino e teste com dados independentes e com o treino e teste efectuados com dados alternados. Nas tabelas 4.13 e 4.14 encontram-se os valores dos erros, correspondentes a cada uma das experiências.

Por fim, no gráfico 4.15 está a relação entre o valor de erro da classificação e o número de épocas utilizadas. Para a construção do gráfico é usado o método de construção das matrizes de treino e teste, que apresenta melhores resultados. Neste caso, mais uma vez, trata-se do método alternado.

Tabela 4.13 Resultados da classificação com OAASVM linear, usando os dados das caras para treino e casas para teste

	Resultado erro classificação (%)	Erro 1ª Iteração (%)	Erro 2ª Iteração (%)	Erro 3ª Iteração (%)	Erro 4ª Iteração (%)	Erro 5ª Iteração (%)
Resultados	30,57	25,02	23,56	22,58	21,27	20,26

Tabela 4.14 Resultados da classificação com OAASVM linear alternado

	Resultado erro classificação (%)	Erro 1ª Iteração (%)	Erro 2ª Iteração (%)	Erro 3ª Iteração (%)	Erro 4ª Iteração (%)	Erro 5ª Iteração (%)
Resultados	26,84	21,17	17,87	16,45	14,52	13,71

Os resultados obtidos para o erro, são bastante inferiores aos obtidos para o classificador MVSVM linear. Se comparar-se os resultados adquiridos com os obtidos pelo mesmo classificador, mas sem recurso à PCA para determinação das características, estes são também bastantes melhores. Por fim verifica-se que o pós-processamento se revelou bastante eficaz.

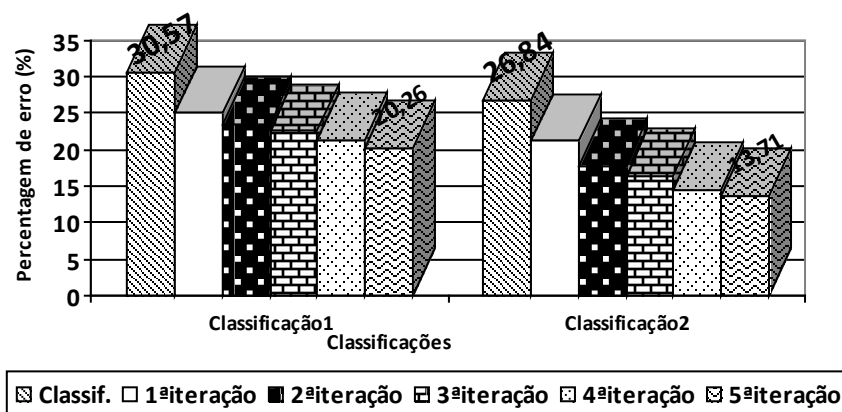


Fig4.14 Gráfico dos resultados da classificação do OAASVM linear

Os resultados mostram que a Classificação2 apresenta melhores resultados. A Classificação2 corresponde ao caso da utilização do classificador OAASVM linear treinado com o método alternado entre dados de caras e cascas. A diferença entre os dois métodos atinge os 3.73% para a classificação e 6.55% para a quinta iteração do pós-processamento. É devido a estes melhores resultados que o estudo seguinte se baseia no método de treino alternado. No gráfico da figura 4.15, faz-se uma análise à variação do erro consoante o número de eventos.

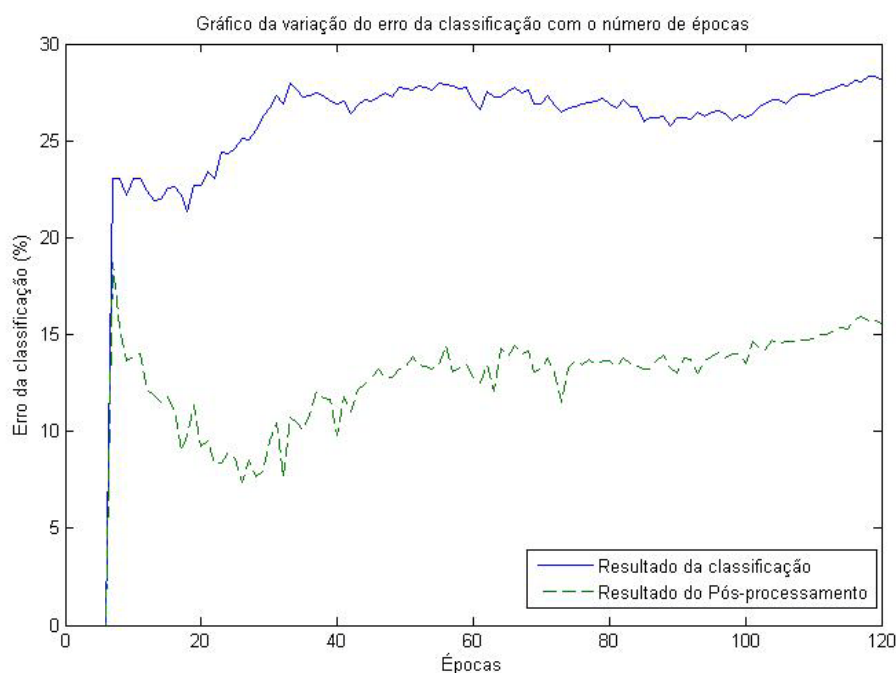


Fig.4.15 Gráfico da variação do erro da classificação OAASVM linear, desde 7 a 120 épocas.

Os resultados da figura 4.15, mostram que o erro da classificação apresenta um mínimo por volta das 20 épocas, aumento ligeiramente a partir desse valor. É visível também o significativo contributo do pós-processamento, uma vez há uma redução considerável do erro. Os resultados da quinta iteração do pós-processamento apresentam o seu mínimo entre 20 e 30 épocas, ligeiramente desfasado do mínimo da classificação inicial.

4.3.3 Classificador OAASVM não linear

Nesta última secção, são expostos os diversos resultados das classificações feitas usando o classificador OAASVM, tal como no subcapítulo anterior, mas na sua vertente não linear. À semelhança dos testes anteriores, também não foi utilizada a PCA na identificação dos vectores de energias característicos a cada indivíduo.

Seguindo a mesma fisionomia dos subcapítulos anteriores, na figura 4.16 apresentam-se os resultados da classificações com treino e teste com dados independentes (Classificação1) e com o treino e teste efectuados com dados alternados (Classificação2). Nas tabelas 4.15 e 4.16 discriminam-se os valores dos erros, correspondentes a cada uma das experiências.

Para finalizar, no gráfico 4.17 encontra-se o gráfico que relaciona o número de épocas, com o erro da classificação e do pós-processamento.

Tabela 4.15 Resultados da classificação com OAASVM não linear, usando os dados das caras para treino e casas para teste

	Resultado erro classificação (%)	Erro 1ª Iteração (%)	Erro 2ª Iteração (%)	Erro 3ª Iteração (%)	Erro 4ª Iteração (%)	Erro 5ª Iteração (%)
Média	26,99	21,54	18,32	16,70	15,16	14,49

Tabela 4.16 Resultados da classificação com OAASVM não linear alternado

	Resultado erro classificação (%)	Erro 1ª Iteração (%)	Erro 2ª Iteração (%)	Erro 3ª Iteração (%)	Erro 4ª Iteração (%)	Erro 5ª Iteração (%)
Média	17,43	12,05	9,78	8,49	6,96	6,62

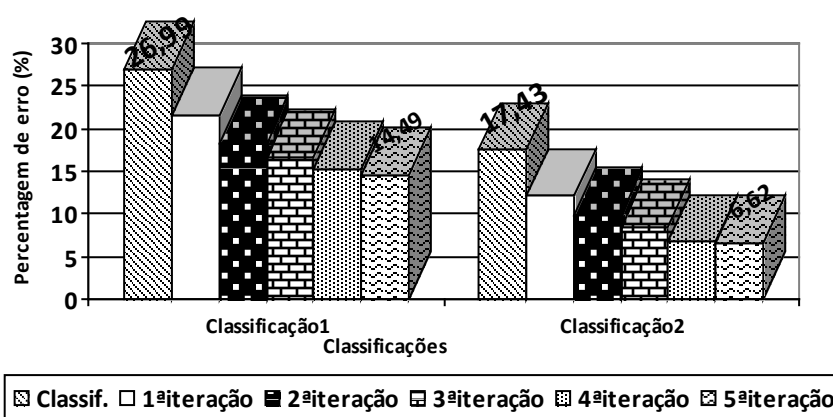


Fig4.16 Gráfico dos resultados da classificação do OAASVM não linear

Quanto aos resultados, comparados com os do classificador MVSVM não linear, estes são bastante parecidos, mas ligeiramente piores no que toca ao erro da classificação sem pós-processamento. Ao comparar-se com os valores de erro deste

mesmo classificador mas aplicado aos dados sujeitos a PCA, os resultados são significativamente melhores.

Os erros obtidos para a classificação com o OAASVM não linear, revelaram-se inferiores quando o classificador é treinado com o método alternado. O pós-processamento, mais uma vez mostra-se eficaz na tarefa de redução do erro, no entanto não existe grande diferença entre a quarta e a quinta iteração.

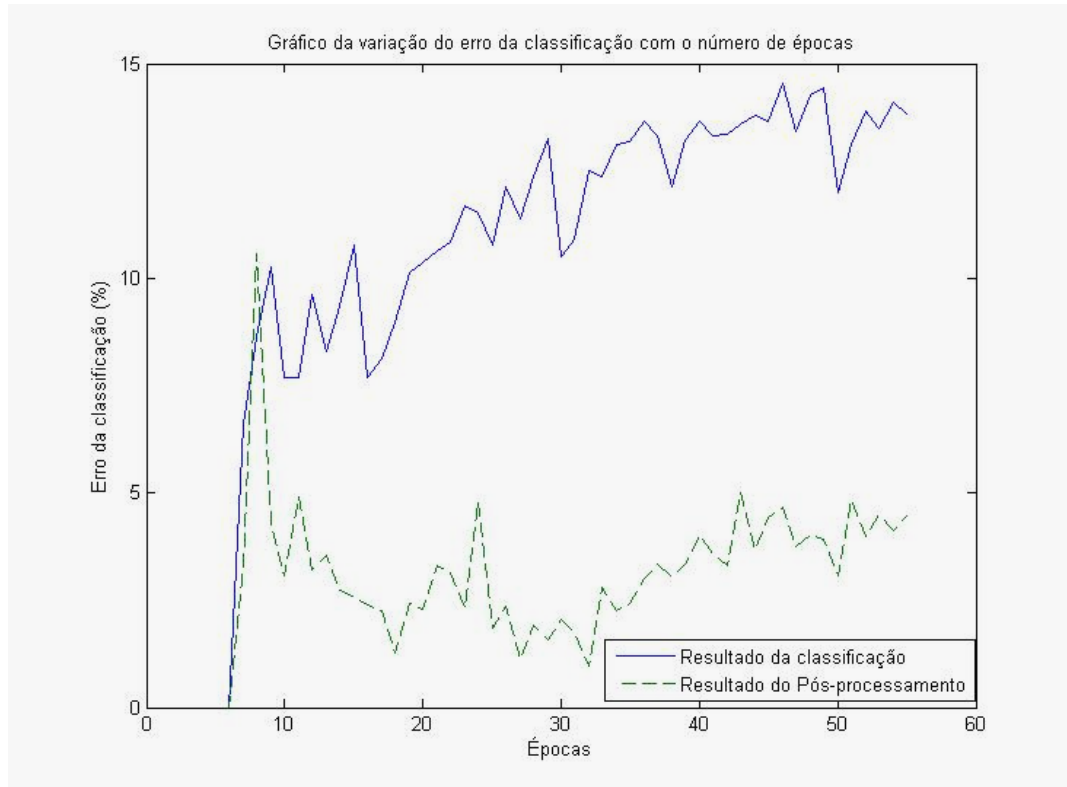


Fig.4.17 Gráfico da variação do erro da classificação OAASVM não linear, desde 7 a 120 épocas.

Tal como o gráfico da figura 4.9, correspondente ao classificador OAASVM não linear com uso de PCA, também foram utilizadas apenas 55 épocas para a construção do gráfico da figura 4.17. As razões pelas quais foram utilizadas 55 épocas são as mesmas apresentadas para o gráfico da figura 4.9.

O gráfico mostra um crescimento do valor do erro da classificação em função do número de épocas. Quanto ao pós-processamento, verifica-se que inicialmente sobrepõe o gráfico da classificação, o que implica um erro superior ao da classificação que se pretende melhorar. Com o aumento do número de épocas, o pós-processamento revela-se bastante eficaz e o valor mínimo de erro encontra-se no intervalo entre 25 e as 35 épocas.

4.4 Resultados finais

Nos subcapítulos anteriores foram apresentados os resultados das classificações de todos os classificadores. Nesta Fase pretende-se reunir esses mesmos resultados e expô-los de modo a tornar evidente qual o classificador mais vantajoso. Foram recolhidos os resultados apenas das classificações alternadas, pois à excepção do classificador OAASVM linear com recurso a PCA para identificação das características, este método de treino é o que apresenta melhores resultados.

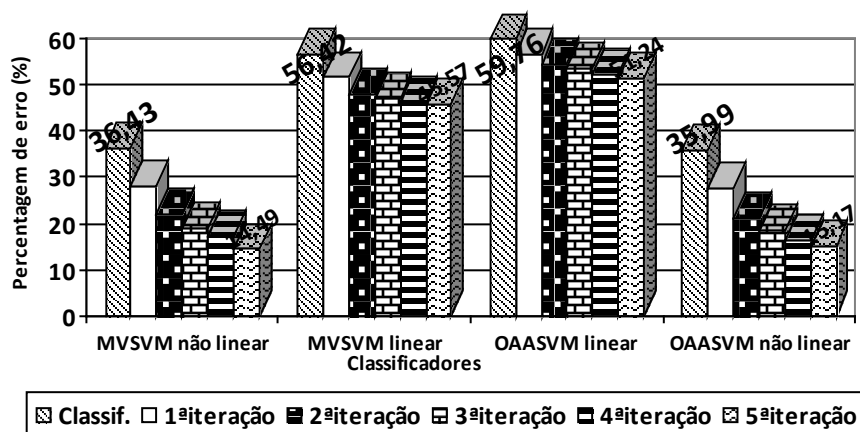


Fig4.18 Gráfico dos resultados dos classificadores com o uso da PCA

O gráfico da figura 4.18 pretende mostrar as diferenças dos valores do erro total das classificações, com o uso dos classificadores enunciados anteriormente. Neste gráfico foram considerados apenas os resultados com PCA. Como é visível, os valores mais elevados de erro acontecem para o classificador OAASVM linear, seguindo-se o MVSVM linear. Os melhores resultados encontram-se entre os classificadores não linear, não existindo grande diferença entre os dois. O OAASVM apresenta melhores valores de classificação, mas após o pós-processamento, o MVSVM não linear apresenta valores inferiores de erro.

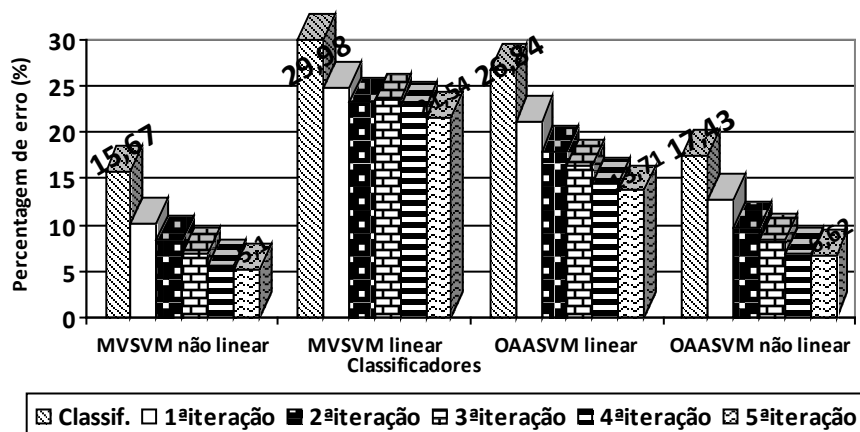


Fig4.19 Gráfico dos resultados dos classificadores sem uso da PCA

Na figura 4.19 encontram-se os resultados dos erros dos classificadores abordados, mas sem o uso da PCA na identificação dos vectores das características. Comparativamente com os resultados com PCA, verifica-se que para todos os casos o erro é inferior sem PCA. Analisando os resultados sem PCA, os melhores valores de erro correspondem ao classificador MVSVM não linear, seguidos do OAASVM não linear. Os

dois classificadores na sua vertente linear apresentam resultados piores com o OAASVM a ser ligeiramente melhor que o MVSVM linear.

Nos dois gráficos anteriores (4.18 e 4.19), verificou-se que o classificador MVSVM apresenta bons resultados e na sua versão não linear encontram-se os melhores resultados obtidos nesta tese.

Surge a necessidade de se analisar ao pormenor os resultados obtidos para o melhor caso, para se tentar perceber como varia o erro por individuo. Foi por esta razão que se construiu a tabela 4.17, onde se apresenta as percentagens das classificações individuais.

Tabela 4.17 Resultados individuais da decisão do classificador MVSVM não linear para 30 épocas e 13 pessoas

		Número dos indivíduos decididos pelo classificador												
Números dos indivíduos que estão a ser alvo de classificação		1	2	3	4	5	6	7	8	9	10	11	12	13
	1	96,6%	3,4%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	2	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	3	0%	0%	80%	3,3%	0%	0%	0%	6,7%	0%	0%	3,3%	3,3%	3,4%
	4	3,4%	0%	0%	86,6%	0%	0%	0%	6,7%	0%	0%	3,3%	0%	0%
	5	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%
	6	13,4%	3,3%	0%	0%	0%	66,7%	0%	3,3%	0%	3,3%	0%	6,7%	3,3%
	7	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%
	8	0%	0%	3,4%	0%	0%	0%	3,3%	93,3%	0%	0%	0%	0%	0%
	9	0%	26,7%	0%	0%	3,3%	0%	0%	0%	70%	0%	0%	0%	0%
	10	0%	6,7%	3,3%	3,4%	0%	0%	3,3%	3,3%	0%	70%	10%	0%	0%
	11	3,4%	3,3%	0%	0%	0%	0%	0%	0%	0%	0%	90%	0%	3,3%
	12	0%	0%	0%	0%	0%	0%	0%	3,4%	0%	0%	0%	93,3%	3,3%
	13	0%	0%	0%	3,3%	0%	0%	0%	6,7%	0%	0%	0%	0%	90%

Para a obtenção dos resultados apresentados na tabela anterior, foram usados 30 épocas para a classificação com o MVSVM não linear sem PCA. A escolha das 30 épocas deve-se ao facto de, como foi demonstrado anteriormente não ser necessário um grande número de épocas de teste para se adquirirem os melhores resultados. De facto na figura 4.11 é visível que o valor mínimo de erro encontra-se por volta das 30 épocas. Foram usadas todos os 13 indivíduos disponíveis nesta classificação.

Analisando a tabela 4.17 é visível que o classificador classifica diferentemente as diferentes pessoas, pois apresenta diversos valores de percentagem de acerto para as os sujeitos de teste. Esta percentagem de acerto encontra-se a na diagonal da matriz (a sombreado) e os outros dados correspondem às percentagens de erro da classificação

individual. Cada linha da tabela pretende representar as 13 classes sujeitas a classificação das 30 épocas. Para o primeiro caso, por exemplo, das 30 épocas por pessoa, o classificador acertou com 96,6% das vezes na pessoa 1 e errou 3,3% ao decidir pela pessoa 2. Podemos verificar que para as pessoas 2 5 e 7, o classificador decidiu sempre bem, e que o pior sujeito foi o individuo 6 com apenas 66,7% de acerto. Neste caso, as percentagens de erro encontram-se distribuídas por vários sujeitos, sendo a mais elevada 13,3% para a pessoa 1.

Os resultados da tabela 4.17 são os resultados do classificador sem pós-processamento. Na tabela 4.18 apresenta-se os resultados da quinta iteração do pós-processamento.

Tabela 4.18 Resultados individuais da decisão do classificador MVSVM não linear para 30 épocas e 13 pessoas após a quinta iteração do pós-processamento

		Número dos indivíduos decididos pelo classificador												
Números dos indivíduos que estão a ser alvo de classificação		1	2	3	4	5	6	7	8	9	10	11	12	13
	1	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	2	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	3	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	4	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	5	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%
	6	3,3%	0%	0%	0%	0%	93,4%	0%	0%	0%	0%	0%	0%	3,3%
	7	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%
	8	0%	0%	0%	0%	0%	0%	6,7%	93,3%	0%	0%	0%	0%	0%
	9	0%	13,3%	0%	0%	0%	0%	0%	0%	86,7%	0%	0%	0%	0%
	10	0%	0%	3,3%	0%	0%	0%	0%	0%	0%	93,4%	3,3%	0%	0%
	11	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%
	12	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%
	13	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%

Da análise das duas tabelas apresentadas para o erro individual, confirma-se uma melhoria significativa nos resultados. A classificação individual melhorou para quase todos os sujeitos, chegando alguns a atingir os 100% de acerto. O único caso onde a percentagem de acerto se manteve foi para o individuo 8. É de destacar que neste caso o pós-processamento errou na sua função pois na tabela 4.17 a percentagem de erro encontra-se distribuída pelos indivíduos 3 e 7, ao passo que, na tabela 4.18 o erro encontra-se concentrado no individuo 7. O pós-processamento decidiu erradamente, que a época que tinha sido atribuída ao sujeito 3, é do sujeito 7 em vez do individuo 8 que seria o correcto. Por fim, é de destacar que o erro geral desta classificação é de 12,56% e após o pós-processamento de 2,56%.

Conclusões

Neste capítulo pretende-se expor as conclusões a que se chegaram nesta tese. As conclusões apresentadas têm por base os resultados obtidos no capítulo 4, para os diversos classificadores e referentes às várias estratégias tomadas para atingir os melhores resultados.

No subcapítulo 4.2.1 desta tese, apresentam-se os resultados referentes à classificação de um classificador (MVSVM não linear), com e sem as épocas onde existiam artefactos. Estas épocas com artefactos foram assinaladas como descartáveis, enquanto se registava o sinal EEG. Os resultados mostram que existe alguma diferença nos resultados, uma vez que o resultado da classificação onde existem artefactos é de 47,47% e sem artefactos 43,93%. Existe uma discrepância de 3,54% entre os dois resultados. Pode-se concluir com base nos resultados que, uma identificação biométrica baseada em PEV apresenta melhores resultados, se não forem consideradas para classificação, épocas com artefactos.

O capítulo dos resultados encontra-se dividido em dois grandes subcapítulos. Esta divisão deve-se ao facto de se pretender apresentar os resultados com e sem PCA para todos os classificadores. O objectivo era compreender a importância do uso da PCA nos dados disponíveis para realizar esta tese. Os resultados mostram que, para todos os casos, uma classificação com o uso da PCA apresenta um valor de erro superior ao caso onde não se recorre à PCA para identificar as energias características individuais. Uma possível explicação, pode estar relacionada com o facto de ter sido utilizado um filtro passa banda, entre os 0.05 e os 30Hz, no processo de aquisição do sinal EEG. Este filtro faz parte do protocolo de aquisição do EEG utilizado. O facto de ter sido utilizada a banda Gama (30-50Hz) para identificação das características, pode estar relacionado com o insucesso do uso da PCA. No gráfico da figura 3.7, mostra que a energia do sinal registado apresenta valores bastante atenuados para as frequências entre os 30 e os 50Hz. A PCA também contribui para uma diminuição da densidade espectral de potência dos canais e é possível que seja por esta razão que os resultados atingidos com PCA sejam piores que os sem PCA.

Foram adoptadas duas estratégias para a construção das matrizes de treino e teste do classificação. A existência de dois métodos, deve-se ao facto de os sinais EEG utilizados nesta tese, terem sido diferenciados em dois conjuntos de sinais. Em um deles, era pedido aos indivíduos que se focassem na diferenciação das caras das pessoas e em outro que se prestassem atenção à alteração das imagens das casas, presentes nos estímulos visuais. Um dos dois métodos usados foi usar épocas de ambas as experiências (metade de cada caso), para treinar e testar o classificador. Chamamos a este método, o método alternado. No outro método foram usadas apenas as épocas das experiências com caras para treino e casas para teste. Estes dois métodos foram testados em todas as classificações e, em todos os casos, foram usadas todas as 13 pessoas para classificação. Os resultados mostram que, em todas as situações, à excepção do OAASVM linear, o método de construção de matrizes alternado é o mais eficaz. Uma possível explicação para estes resultados, pode residir no facto de, ser pedido aos indivíduos que se foquem em estímulos diferentes, num dos casos caras e num outro casas. Apesar de os estímulos visuais não serem muito diferentes uns dos outros e, em ambos os casos se registarem PEV, o processamento cerebral que é pedido, é ligeiramente diferente. Como no caso alternado o classificador é treinado com dados das duas experiências, o classificador responde melhor quando testado com

energias dos dois casos, pois as fronteiras das classes já prevêm as duas situações. O facto de se verificar que o método alternado não é o melhor para o classificador OAASVM linear, pode-se dever ao facto de o erro do classificador, pelo facto de ser linear, não conseguir definir devidamente as melhores fronteiras para as 13 classes. O resultado é um erro bastante grande, de 59,76% (erro da classificação), que é o maior erro obtido, quando comparado com todas as outras classificações.

Para a classificação foram usados dois classificadores, MVSVM e o OAASVM. Em ambos os casos foram usados os classificadores nas suas vertentes lineares e não lineares. O que se pretendia era descobrir se existia uma diferença significativa entre os dois casos. Os resultados mostram que os classificadores não lineares são mais eficazes, apresentado um erro significativamente inferior. A justificação para este facto encontra-se relacionada com o próprio processo de treino do classificador. O classificador SVM, é um classificador binário e é necessário fazer-se uma divisão entre os dados de duas classes. Quando os dados são linearmente separáveis, um classificador linear não terá qualquer dificuldade em proceder a esta divisão. No entanto muito raramente isto acontece e quando os dados não são linearmente separáveis, é necessário introduzir um erro calculado, para proceder ao treino do classificador. Com o classificador não linear, isto não se verifica (como é devidamente explicado no capítulo 2 na secção 2.2.3), sendo a classificação mais eficaz.

Como foi referido, foram usadas duas vertentes do classificador SVM, o MVSVM e o OAASVM. Estes classificadores são baseados no SVM, mas têm a vantagem de possibilitarem muticlasse, uma vez que o SVM é um classificador binário. Os resultados mostram que não existem grandes diferenças nos resultados obtidos entre os dois classificadores, no entanto existe uma pequena vantagem para o MVSVM. Os melhores resultados são obtidos com o classificador MVSVM e para além disso, como utiliza uma estratégia de treino para multiclasse, *one against other* (OAO), contrariamente ao OAASVM (*one against all* SVM), é computacionalmente mais viável. Os menores valores de erro são conseguidos para o classificador MVSVM não linear sem PCA, batendo o OAASVM nas mesmas condições por 1,76%. Por estas razões considera-se o MVSVM o melhor classificador para classificar as características individuais.

Para a implementação de um sistema de identificação viável, implica também conhecer o valor do número de épocas mínimo que possibilite uma boa classificação. Para se conhecer este valor, recorreu-se a uma análise de gráficos do valor do erro em função do número de épocas de teste. Foi possível verificar, que existe uma melhor resposta por parte dos classificadores não lineares do que os lineares, uma vez que, os não lineares apresentam um aumento do erro menos acentuado, á medida que se aumentam as épocas. Apesar deste aumento, para os classificadores não lineares, verifica-se que existe um valor mínimo de erro, entre as 20 e as 35 épocas. Como cada época corresponde a uma porção de 1,6 segundos de sinal EEG, se não se considerar as épocas com artefactos e pequenas porções de sinal entre as épocas, para o teste do classificador seriam necessários entre 32 a 56 segundos de sinal EEG.

Quanto ao pós-processamento (PP), este revelou-se, de um modo geral, bastante eficaz na sua função de redução do erro. É de destacar o melhor comportamento do PP quando o valor do erro da classificação não é muito grande. Quando o erro ultrapassa os 50%, verifica-se que o PP não consegue reduzir o erro tão significativamente. Isto acontece pois o PP tem a função de suavização dos dados, o que não é possível se no vector com as previsões do classificador, não se verificarem sequências correctas de decisões. Por vezes verifica-se que o PP introduz erro em vez de o remover, isto acontece quando a classificação já apresenta um erro bastante elevado, ou em alguns

casos, quando se testa o classificador com poucas épocas. De um modo geral o pós-processamento reduz o erro e contribui para uma melhor identificação biométrica.

Os melhores resultados, conseguidos para o classificador MVSVM não linear (sem PCA), com erro de pós-processamento de 5,1%, mostram que, apesar da atenuação causada pelo filtro (0.05-30Hz) no processo de aquisição do EEG e de terem sido usados diferentes estímulos visuais, é possível conseguir-se uma boa classificação.

Estes resultados mostram que foram cumpridos os objectivos desta tese. É possível criar-se um sistema biométrico baseado em sinais EEG, como o descrito nesta tese. Nesta tese, no processo de classificação, é necessário que sejam introduzidos vários vectores das características de determinado indivíduo consecutivamente, de modo a tornar possível um processo de pós-processamento e deste modo melhorar os erros da classificação.

Referências

- [1] Pankanti, S.,Prabh akar,S.,and Jain,A.K.: ‘On the individuality of fingerprints’, IEEE Trans. Pattern Anal. Machine Intell.,2002, 24,(8) , pp. 1010–1025
- [2] A. K. Jain, A. Ross and S. Pankanti, “A prototype hand geometry-based verification system,” *Proc. 2nd International Conference on Audio and Video-Based Biometric Person Identification*, 1999, pp 166-171.
- [3] P. Verlinde, G. Chollet and M. Acheroy, “Multi-modal identity verification using expert fusion,” *Information Fusion*, vol. 1, pp. 17-33, 2000
- [4] M. Poulos, M. Rangoussi, V. Chrissicopoulos and A. Evangelou, “Person identification based on parametric processing on the EEG,” *Proceedings of the Sixth International Conference on Electronics, Circuits and Systems*, vol. 1, pp. 283-286, 1999
- [5] R. B. Paranjape, J. Mahovsky, L. Benedicenti and Z. Koles, “The electroencephalogram as a biometric,” *Proc. Canadian Conf. Electrical and Computer Engineering*, vol. 2, 2001, pp. 1363-1366.
- [6] R. Palaniappan. Method of identifying individuals using VEP signals and neural network. IEE Proceedings - Science Measurement and Technology, 151(1):16–20, 2004.
- [7] Palaniappan, R.,Rav eendran,P.,and Omatu,S.: ‘VEP optimal channel selection using genetic algorithm for neural network classification of alcoholics’, IEEE Trans. Neural Netw.,2002 , 13,(2) , pp. 486–491
- [8] Basar,E ., Eroglu,C.B.,K arakas,S., and Schurman,M.: ‘Oscillatory brain theory: a new trend in neuroscience’, IEEE Eng. Med. Biol. Mag.,1999, 18,(3), pp. 56–66
- [9] Tallon-Baudry,C.,Bertrand,O., Delpuech,C.,and Pernier,J.: ‘Stimulus specificity of phased-locked and non-phase locked 40Hz visual responses in human’, J. Neurosci.,1996, 16,(13),pp. 4240–4249
- [10] Tallon-Baudry,C.,Bert rand,O., Peronnet, F.,and Pernier,J.: ‘Induced g-band activity during the delay of a visual short-term memory task in humans’, J. Neurosci.,1998, 18,(11), pp. 4244–4254
- [11] K.V.R. Ravi, *Ramaswamy Palaniappan Recognising Individuals Using Their Brain Patterns *Faculty of Information Science and Technology, Multimedia University, Melaka, Malaysia* *Dept of Computer Science, University of Essex, Colchester, United Kingdom, 2005
- [12] KVR. Ravi and Ramaswamy Palaniappan Leave-one-out Authentication of Persons Using 40 Hz EEG Oscillations *Faculty of Information Science and Technology, Multimedia University, Melaka, Malaysia* *Dept of Computer Science, University of Essex, Colchester, United Kingdom, 2005
- [13] Mi Islita, Information Retrieval Intelligence, 2006, <http://www.miislita.com/information-retrieval-tutorial/matrix-tutorial-3-eigenvalues-eigenvectors.html> (acedido em Setembro 2009)

- [14] Shih-Chung Chen Shih-Chang Hsieh Chih-Kuo Liang , An Intelligent Brain Computer Interface of Visual Evoked Potential EEG, *Institute of EE, Southern Taiwan University*, 2008
- [15] Inf. Sci. & Eng. Coll., Northeastern Univ., Shenyang ,EEG Analysis of Alcoholics and Controls Based on Feature Extraction, *Beijing*, 2006
- [16] Brainmaster Technologies, 2009, www.brainmaster.com (Acedido em Agosto de 2009)
- [17] Isabel M. Santos, Jaime Iglesias, Ela I. Olivares e Andrew W. Young, Differential Edeects of object-based attention on evokated potenciales to fearful and disgusted faces, *Universidade de Aveiro, departamento de ciências da Educação, Universidad Autónoma de Madrid, Falcultad de Psicología, Departamento de Psicología Biológica y de la salud, University of York, Departement of Psychology and York Neurolmaging center*, 2007
- [18] Pang-Ning Tan, Michael Steinbach e Vipin Kumar, Introduction to Data Mining, *Addison Wesley*, 2006
- [19] Lindsay I Smith, A tutorial on Principal Components Analysis, 2002

Anexos

Para a identificação das características individuais, foram criadas algumas funções. Este processo de encontrar os vectores das energias específicos para cada indivíduo é conseguido através da seguinte função:

```
%% Função que invoca todo o processo de identificação de características
% para determinado individuo
%%
% [energ_t]=ident_total_final(DATA,EEG,per,A,B)
% Parâmetros de entrada: DATA em formato .wpl, exemplo 'anabffa.wpl'
% Parâmetros de entrada: EEG, matriz com os dados EEG carregados a partir
% do EEGLAB, exemplo EEG.data
% Parâmetros de entrada: per percentagem de eigenvalues de aceitação para
% a reconstrução da matriz de dados
% Parâmetros de entrada:A e B, frequências limite para a filtragem, neste
% caso A=30 e B=50 (Hz)
% Parâmetros de saída:energ_t, é o vector linha das energias de
% determinado sujeito, com dimensão [1x22]
function [energ_t]=ident_total_final(DATA,EEG,per,A,B)

[sinal,lenght]=rejeicao(DATA, EEG);
energ=zeros(lenght,20);
for i=1:lenght

    [W,D,Z1,m]=PCA(sinal, i);
    [reconst]=Eigenvalues(per, D, W, Z1,m);
    [energ]=pwelchfiltro_norm(A,B,reconst);
    energ_t(i,:)=energ;
end
end
```

A função anterior evoca outras funções, a primeira é a rejeição

```
%% leitura do sinal e rejeição das épocas
%%
% Parâmetros de entrada:DATA e um fich .wpl por exemplo anabffa.wpl entre
% pelicas
% Parâmetros de entrada:EEG corresponde a matriz 22x400x251, do sinal
% original do EEG
% Parâmetros de saída:sinal corresponde ao sinal EEG mas sem as épocas
% com
% artefacto
% Parâmetros de saída: Corresponde ao número de épocas que o sinal
%rejeitado possui
function [sinal,lenght]=rejeicao(DATA, EEG_s)

[total,inrej]=fwplread(DATA);
signal=EEG_s;
[a,b]=size(inrej);
lenght=total-b;
sinal=zeros(22,400,lenght);k=1;i=1;j=1;
```

```

while j <= b
    if i == inrej(1,j)
        j=j+1;
        i=i+1;
    else
        sinal(:, :, k)=signal(:, :, i);
        k=k+1;
        i=i+1;
    end
end
for xx=i:total
    sinal(:, :, k)=signal(:, :, i);
    k=k+1;
    i=i+1;
end
end

```

A rejeição por sua vez evoca a função fwplread:

```

function [total,inrej]=fwplread(name1,name2)
%[total,inrej]=fwplread(name1,name2)
% usage
% [t,v]=fwplread('nome.wpl') ou
% [t,v]=fwplread('nome.wpl', 'aaa')---dados em ficheiro mat... c,u,l
% Reading ficheiros *.wpl
%
fid=fopen(name1);
k=1;
[dummy,flag]=fread(fid,1,'char');
while flag
    c(k)=dummy;
    [l(k),flag]=fread(fid,1,'int');
    % Sem confirmação de boa leitura...
    [u(k),flag]=fread(fid,1,'int');
    % sem confirmação de boa leitura.....
    k=k+1; flag=0;
    [dummy,flag]=fread(fid,1,'char');
end
total=k-1;
% indice dos rejeitados parece ser o 3..... se calhar deve ser confirmado
inrej=find(c==3);
fclose(fid)
% ESCRIVER EM FICHEIRO MAT as variáveis lidas?
if nargin==2
    save(name2 , 'c', 'l', 'u');
end

```

A função ident_total_final, chama também a função PCA:

```

%PCA-Principal component analysis
%%
%
%Parâmetro de entrada:Evento,evento que se esta a analisar
%Parâmetro de entrada:sinal, EEG corresponde a matriz 22x400xeventos,

```

```

%do sinal do EEG já sem as épocas rejeitadas
% Parâmetro de saída: W e D, matrizes dos eigenvectors e eigenvalues
% respectivamente
% Parâmetros de saída: Z1 é a matriz do sinal da época
%selecionada[22x400]com
% média (m) nula
function [W,D,Z1,m]=PCA(sinal, Evento)

EEG=sinal;
n=zeros(20,400);
m=zeros(20,1);
for i=1:20
    for j=1:400
        n(i,j)=EEG(i,j,Evento);
    end
    m(i,1)=mean(n(i,:));
end
Z= n;
Z1= Z - m*ones(1,400);
C=(Z1*Z1')./400;
[W,D,O]=svd(C);      %D encontra-s em diagonal decrescente

end

```

A ident_total_final chama Aida a função Eigenvalues:

```

%% Eigenvalues
%%
% W, matriz de eigenvectors
% D, matriz diagonal decrescente dos eigenvalues
% per, percentagem de aceitação de eigenvalues
% Z1, matriz do sinal de um evento com media zero
% m, média do sinal que foi removida na função PCA
% reconst, matriz do sinal de uma época reconstituída depois de removidos
% alguns eigenvectors
function [reconst]=Eigenvalues(per, D, W, Z1,m)

soma=zeros(20,1);
soma= sum(D,2);      %20x1
total=sum(soma,1);
stop=total*per;
parar=0;R=0;i=1;
while parar <= stop
    parar=D(i,i)+parar;
    R=R+1;
    i=i+1;
end
W1=zeros(20,R);      %W1->20xR
%Copia de w para w1 mas so as R colunas
for i=1:20
    for j=1:R
        W1(i,j)=W(i,j);
    end
end
y=W1'*Z1;
%%

```

```
%reconstituição da matriz 20X400
reconst=(W1*y)+m*ones(1,400);
end
```

Por fim, na função `ident_total_final` é chamada a função `pwelchfiltro_norm`:

```
% Pwelch utilizada como filtro e para calcular as energias de cada canal
%e
% posteriormente normaliza-las
% A função devolve energias por evento
% A e B representam o intervalo de frequências que é utilizado como passa
% banda
% reconst é o sinal reconstituído após a PCA e remoção de alguns
% eigenvectors
function [energ]=pwelchfiltro_norm(A,B,reconst)

Fs = 250; %frequencia de amostragem
energ=zeros(1,20);
for j=1:20

    [xx,f]=pwelch(reconst(j,:),[],[],[],Fs); % Uses default window,
    overlap & NFFT.
    [tamanho,x]=size(f);
    for i=1:tamanho
        if f(i)>=A && f(i)<=B
            energ(j)= xx(i)+energ(j);
        end
    end
end
total=sum(energ);
energ=energ./total;
end
```

Depois de construídas as bases de dados com os vectores das características individuais, são utilizadas funções para classificação como a `func_class_mix_svm`. Esta função é a responsável pela classificação quando se constroem as matriz de treino e teste com o modo alternado.

```
% função de classificação pessoal usando o classificador SVM
%function[erro]=func_class_mix_svm(carateristicas,n_pessoas,eve_var,tipos)
%carateristicas é o ficheiro .mat com todas as energias dos sujeitos, se
%caracteristicas for 10, significa que é com PCA, 11 é sem PCA
%n_pessoas é o numero de pessoas que se pretende usar para a
classificacao
%eve_var é o numero de épocas que s pretende usar para treino e teste por
%pessoa
%tipo é a variavel que define o classificador usado.
% Se tipo for: 1- é o mvsvm nao linear;
% 2- mvsvm linear; 3- oaasvm nao linear e 4- oaasvm linear
% Como parâmetros de saída tem-se o erro_inicial, que é o erro da
% classificação; ypred, que é o vector com as previsoes do classificador
% tst_necessario é um vector com os lables correctos para testar a
```

```

% classificaça

function[erro_inicial,ypred,tst_necessario]=func_class_mix_svm(caract,n_p
essoas,eve_var,tipo)

%% carregar dados
if caract==11
    load caracteristicas11;
end
if caract==10
    load caracteristicas10;
end
if caract~= 10 && caract~=11
    display('Erro, parametro de entrada caract diferente de 10 ou 11')
end
%% criar os vectores para guardar as energias
energ_t_anabffa_20=zeros(222,20);energ_t_anabhfa_20=zeros(201,20);
energ_t_anagffa_20=zeros(108,20);energ_t_anaghfa_20=zeros(152,20);
energ_t_emmaffb_20=zeros(148,20);energ_t_emmahfb_20=zeros(109,20);
energ_t_encaffa_20=zeros(150,20);energ_t_encahfa_20=zeros(153,20);
energ_t_evalffa_20=zeros(177,20);energ_t_evalhfa_20=zeros(220,20);
energ_t_guacffa_20=zeros(131,20);energ_t_guachfa_20=zeros(92,20);
energ_t_laugffa_20=zeros(172,20);energ_t_laughfa_20=zeros(114,20);
energ_t_loraffb_20=zeros(148,20);energ_t_lorahfb_20=zeros(194,20);
energ_t_marchfb_20=zeros(91,20);energ_t_marcffb_20=zeros(150,20);
energ_t_mcemhfb_20=zeros(154,20);energ_t_mcemffb_20=zeros(166,20);
energ_t_noemffb_20=zeros(150,20);energ_t_noemhfb_20=zeros(120,20);
energ_t_patvhfa_20=zeros(156,20);energ_t_patvffa_20=zeros(171,20);
energ_t_yolbhfa_20=zeros(220,20);energ_t_yolbffa_20=zeros(208,20);
%% Retirar os canais oculares
for i=1:20
    energ_t_anabffa_20(:,i)=energ_t_anabffa(:,i);
    energ_t_anabhfa_20(:,i)=energ_t_anabhfa(:,i);
    energ_t_anagffa_20(:,i)=energ_t_anagffa(:,i);
    energ_t_anaghfa_20(:,i)=energ_t_anaghfa(:,i);
    energ_t_emmaffb_20(:,i)=energ_t_emmaffb(:,i);
    energ_t_emmahfb_20(:,i)=energ_t_emmahfb(:,i);
    energ_t_encaffa_20(:,i)=energ_t_encaffa(:,i);
    energ_t_encahfa_20(:,i)=energ_t_encahfa(:,i);
    energ_t_evalffa_20(:,i)=energ_t_evalffa(:,i);
    energ_t_evalhfa_20(:,i)=energ_t_evalhfa(:,i);
    energ_t_guacffa_20(:,i)=energ_t_guacffa(:,i);
    energ_t_guachfa_20(:,i)=energ_t_guachfa(:,i);
    energ_t_laugffa_20(:,i)=energ_t_laugffa(:,i);
    energ_t_laughfa_20(:,i)=energ_t_laughfa(:,i);
    energ_t_loraffb_20(:,i)=energ_t_loraffb(:,i);
    energ_t_lorahfb_20(:,i)=energ_t_lorahfb(:,i);
    energ_t_marcffb_20(:,i)=energ_t_marcffb(:,i);
    energ_t_marchfb_20(:,i)=energ_t_marchfb(:,i);
    energ_t_mcemffb_20(:,i)=energ_t_mcemffb(:,i);
    energ_t_mcemhfb_20(:,i)=energ_t_mcemhfb(:,i);
    energ_t_noemffb_20(:,i)=energ_t_noemffb(:,i);
    energ_t_noemhfb_20(:,i)=energ_t_noemhfb(:,i);
    energ_t_patvffa_20(:,i)=energ_t_patvffa(:,i);
    energ_t_patvhfa_20(:,i)=energ_t_patvhfa(:,i);
    energ_t_yolbffa_20(:,i)=energ_t_yolbffa(:,i);
    energ_t_yolbhfa_20(:,i)=energ_t_yolbhfa(:,i);
end

```



```

%%
tamanho=zeros(1,26);
[tamanho(1,1),x]=size(energ_t_anabffa_20);[tamanho(1,2),x]=size(energ_t_a
nabhfa_20);
[tamanho(1,3),x]=size(energ_t_anagffa_20);[tamanho(1,4),x]=size(energ_t_a
naghfa_20);
[tamanho(1,5),x]=size(energ_t_emmafffb_20);[tamanho(1,6),x]=size(energ_t_e
mmahfb_20);
[tamanho(1,7),x]=size(energ_t_encaffa_20);[tamanho(1,8),x]=size(energ_t_e
ncahfa_20);
[tamanho(1,9),x]=size(energ_t_evalffa_20);[tamanho(1,10),x]=size(energ_t_
evalhfa_20);
[tamanho(1,11),x]=size(energ_t_guacffa_20);[tamanho(1,12),x]=size(energ_t
_guachfa_20);
[tamanho(1,13),x]=size( energ_t_laugffa_20);[tamanho(1,14),x]=size(
energ_t_laughfa_20);
[tamanho(1,15),x]=size(energ_t_lorafffb_20);[tamanho(1,16),x]=size(energ_t
_lorahfb_20);
[tamanho(1,17),x]=size(energ_t_marcfffb_20);[tamanho(1,18),x]=size(energ_t
_marchfb_20);
[tamanho(1,19),x]=size(energ_t_mcemfffb_20);[tamanho(1,20),x]=size(energ_t
_mcemhfb_20);
[tamanho(1,21),x]=size(energ_t_noemfffb_20);[tamanho(1,22),x]=size(energ_t
_noemhfb_20);
[tamanho(1,23),x]=size(energ_t_patvffa_20);[tamanho(1,24),x]=size(energ_t
_patvhfa_20);
[tamanho(1,25),x]=size(energ_t_yolbffa_20);[tamanho(1,26),x]=size(energ_t
_yolbhfa_20);
eventos=sum(tamanho);
%%
eventos_trn=0;
for i=1:1:26
    eventos_trn=eventos_trn+floor(tamanho(1,i)/2);
end
eventos_tst=0;
for i=1:1:26
    eventos_tst=eventos_tst+round(tamanho(1,i)/2);
end
dados_tst=zeros(20,eventos_tst);
dados_trn=zeros(20,eventos_trn);
k=1;
for i=1:tamanho(1,1)/2
    dados_trn(:,k)=energ_t_anabffa_20(i,:);
    k=k+1;
end
for i=1:floor(tamanho(1,2)/2)
    dados_trn(:,k)=energ_t_anabhfa_20(i,:);
    k=k+1;
end
for i=1:tamanho(1,3)/2
    dados_trn(:,k)=energ_t_anagffa_20(i,:);
    k=k+1;
end
for i=1:tamanho(1,4)/2
    dados_trn(:,k)=energ_t_anaghfa_20(i,:);
    k=k+1;
end
for i=1:tamanho(1,5)/2
    dados_trn(:,k)=energ_t_emmafffb_20(i,:);

```

```

        k=k+1;
    end
    for i=1:floor(tamanho(1,6)/2)
        dados_trn(:,k)=energ_t_emmahfb_20(i,:);
        k=k+1;
    end
    for i=1:tamanho(1,7)/2
        dados_trn(:,k)=energ_t_encaffa_20(i,:);
        k=k+1;
    end
    for i=1:floor(tamanho(1,8)/2)
        dados_trn(:,k)=energ_t_encahfa_20(i,:);
        k=k+1;
    end
    for i=1:floor(tamanho(1,9)/2)
        dados_trn(:,k)=energ_t_evalffa_20(i,:);
        k=k+1;
    end
    for i=1:tamanho(1,10)/2
        dados_trn(:,k)=energ_t_evalhfa_20(i,:);
        k=k+1;
    end
    for i=1:floor(tamanho(1,11)/2)
        dados_trn(:,k)=energ_t_guacffa_20(i,:);
        k=k+1;
    end
    for i=1:tamanho(1,12)/2
        dados_trn(:,k)=energ_t_guachfa_20(i,:);
        k=k+1;
    end
    for i=1:tamanho(1,13)/2
        dados_trn(:,k)=energ_t_laugffa_20(i,:);
        k=k+1;
    end
    for i=1:tamanho(1,14)/2
        dados_trn(:,k)=energ_t_laughfa_20(i,:);
        k=k+1;
    end
    for i=1:tamanho(1,15)/2
        dados_trn(:,k)=energ_t_loraffb_20(i,:);
        k=k+1;
    end
    for i=1:tamanho(1,16)/2
        dados_trn(:,k)=energ_t_lorahfb_20(i,:);
        k=k+1;
    end
    for i=1:tamanho(1,17)/2
        dados_trn(:,k)=energ_t_marcffb_20(i,:);
        k=k+1;
    end
    for i=1:floor(tamanho(1,18)/2)
        dados_trn(:,k)=energ_t_marchfb_20(i,:);
        k=k+1;
    end
    for i=1:tamanho(1,19)/2
        dados_trn(:,k)=energ_t_mcemffb_20(i,:);
        k=k+1;
    end
    for i=1:tamanho(1,20)/2

```

```

        dados_trn(:,k)=energ_t_mcemhfb_20(i,:);
        k=k+1;
    end
    for i=1:tamanho(1,21)/2
        dados_trn(:,k)=energ_t_noemffb_20(i,:);
        k=k+1;
    end
    for i=1:tamanho(1,22)/2
        dados_trn(:,k)=energ_t_noemhfb_20(i,:);
        k=k+1;
    end
    for i=1:floor(tamanho(1,23)/2)
        dados_trn(:,k)=energ_t_patvffa_20(i,:);
        k=k+1;
    end
    for i=1:tamanho(1,24)/2
        dados_trn(:,k)=energ_t_patvhfa_20(i,:);
        k=k+1;
    end
    for i=1:tamanho(1,25)/2
        dados_trn(:,k)=energ_t_yolbffa_20(i,:);
        k=k+1;
    end
    for i=1:tamanho(1,26)/2
        dados_trn(:,k)=energ_t_yolbhfa_20(i,:);
        k=k+1;
    end
    %%
    k=1;
    for i=(tamanho(1,1)/2+1):tamanho(1,1)
        dados_tst(:,k)=energ_t_anabffa_20(i,:);
        k=k+1;
    end
    for i=round(tamanho(1,2)/2):tamanho(1,2)
        dados_tst(:,k)=energ_t_anabhfa_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,3)/2+1):tamanho(1,3)
        dados_tst(:,k)=energ_t_anagffa_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,4)/2+1):tamanho(1,4)
        dados_tst(:,k)=energ_t_anaghfa_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,5)/2+1):tamanho(1,5)
        dados_tst(:,k)=energ_t_emmaffb_20(i,:);
        k=k+1;
    end
    for i=round(tamanho(1,6)/2):tamanho(1,6)
        dados_tst(:,k)=energ_t_emmahfb_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,7)/2+1):tamanho(1,7)
        dados_tst(:,k)=energ_t_encaffa_20(i,:);
        k=k+1;
    end
    for i=round(tamanho(1,8)/2):tamanho(1,8)
        dados_tst(:,k)=energ_t_encahfa_20(i,:);

```

```

        k=k+1;
    end
    for i=round(tamanho(1,9)/2):tamanho(1,9)
        dados_tst(:,k)=energ_t_evalffa_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,10)/2+1):tamanho(1,10)
        dados_tst(:,k)=energ_t_evalhfa_20(i,:);
        k=k+1;
    end
    for i=round(tamanho(11)/2):tamanho(11)
        dados_tst(:,k)=energ_t_guacffa_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,12)/2+1):tamanho(1,12)
        dados_tst(:,k)=energ_t_guachfa_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,13)/2+1):tamanho(1,13)
        dados_tst(:,k)=energ_t_laugffa_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,14)/2+1):tamanho(1,14)
        dados_tst(:,k)=energ_t_laughfa_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,15)/2+1):tamanho(1,15)
        dados_tst(:,k)=energ_t_loraffb_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,16)/2+1):tamanho(1,16)
        dados_tst(:,k)=energ_t_lorahfb_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,17)/2+1):tamanho(1,17)
        dados_tst(:,k)=energ_t_marcffb_20(i,:);
        k=k+1;
    end
    for i=round(tamanho(1,18)/2):tamanho(1,18)
        dados_tst(:,k)=energ_t_marchfb_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,19)/2+1):tamanho(1,19)
        dados_tst(:,k)=energ_t_mcemffb_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,20)/2+1):tamanho(1,20)
        dados_tst(:,k)=energ_t_mcemhfb_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,21)/2+1):tamanho(1,21)
        dados_tst(:,k)=energ_t_noemffb_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,22)/2+1):tamanho(1,22)
        dados_tst(:,k)=energ_t_noemhfb_20(i,:);
        k=k+1;
    end
    for i=round(tamanho(1,23)/2):tamanho(1,23)

```

```

        dados_tst(:,k)=energ_t_patvffa_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,24)/2+1):tamanho(1,24)
        dados_tst(:,k)=energ_t_patvhfa_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,25)/2+1):tamanho(1,25)
        dados_tst(:,k)=energ_t_yolbffa_20(i,:);
        k=k+1;
    end
    for i=(tamanho(1,26)/2+1):tamanho(1,26)
        dados_tst(:,k)=energ_t_yolbhfa_20(i,:);
        k=k+1;
    end
    %%
    eve_trn=0;
    for i=1:1:2*n_pessoas
        eve_trn=eve_trn+floor(tamanho(1,i)/2);
    end
    eve_tst=0;
    for i=1:1:2*n_pessoas
        eve_tst=eve_tst+round(tamanho(1,i)/2);
    end
    ppl=zeros(20,eve_trn);
    k=1;
    for i=1:eve_trn
        ppl(:,i)=dados_trn(:,k);
        k=k+1;
    end
    trn.X=ppl;
    v=ones(1,eve_trn);
    long=zeros(1,26);
    for i=1:1:26
        long(1,i)=floor(tamanho(1,i)/2);
    end
    y=1;i=1;j=1;ev_count=0;
    while ev_count < eve_trn
        for k=1:long(1,i)
            v(1,y)=j*v(1,y);
            y=y+1;
        end
        ev_count=ev_count+long(1,i);
        i=i+1;
        j=round(i/2);
    end
    trn.y=v;
    trn.num_data=eve_trn;
    trn.dim=2;
    trn.name='estrutura_treino';
    %%
    ppl=zeros(20,eve_var*n_pessoas);
    long=zeros(1,26);
    k=1;
    for i=1:1:26
        long(1,k)=round(tamanho(1,i)/2);
        k=k+1;
    end
    y=1;p=2;i=1;ev_count=0;

```

```

while ev_count < eve_var*n_pessoas
    ev_count2=0;
    while ev_count2 < eve_var
        ppl(:,y)=dados_tst(:,i);
        y=y+1;
        ev_count2=ev_count2+1;
        i=i+1;
    end
    i=i+(long(1,p)+long(1,p-1)-eve_var);
    p=p+2;
    ev_count=ev_count+eve_var;
end
tst.X=ppl;
%%
v=ones(1,eve_var*n_pessoas);
j=1;i=1;y=1;ev_count=0;
while ev_count < eve_var*n_pessoas
    for k=1:eve_var
        v(1,y)=i*v(1,y);
        y=y+1;
    end
    ev_count=ev_count+eve_var;
    i=i+1;
end
tst.y=v;
tst.num_data=eve_var*n_pessoas;
tst.dim=2;
tst.name='estrutura_teste';
%%
if tipo==1
    options = struct('ker','rbf','arg',0.1,'C',10,'verb',1);
    model = oaosvm(trn,options);
    [ypred,votes] = mvsvmclass(tst.X,model);
    erro_inicial = cerror(ypred,tst.y);
    tst_necessario=tst.y;
end
if tipo==2
    options = struct('ker','linear','C',10,'verb',1);
    model = oaosvm(trn,options);
    [ypred,votes] = mvsvmclass(tst.X,model);
    erro_inicial = cerror(ypred,tst.y);
    tst_necessario=tst.y;
end
if tipo==3
    options = struct('ker','rbf','arg',0.1,'C',10,'verb',1);
    model = oaosvm(trn,options);
    ypred = svmclass( tst.X, model );
    erro_inicial=ceerror(ypred, tst.y );
    tst_necessario=tst.y;
end
if tipo==4
    options = struct('ker','linear','C',10,'verb',1);
    model = oaosvm(trn,options);
    ypred = svmclass( tst.X, model );
    erro_inicial=ceerror(ypred, tst.y );
    tst_necessario=tst.y;
end
end
end

```

Por fim, apresenta-se a função responsável pelo pós-processamento. O objectivo com esta função é reduzir o erro inicial da classificação.

```
%% Função de Pós-Processamento
%%
% Como parâmetros de entrada tem-se: ypred, que é o vector com os lables
% das previsões do classificador; tst_necessario é um vector com os
% lables correctos para testar a
% classificação
% Para parâmetros de saída tem-se os erros que vão desde erro1...erro5,
% que
% representam o valor do erro após cada iteração do processamento; e os
% parâmetros ypred1..ypred5 representam as previsões do pós processamento
% após cada iteração
function[erro1,erro2,erro3,erro4,erro5,ypred1,ypred2,ypred3,ypred4,ypred5
] = pos_proc(ypred,tst_necessario)

%% para eliminar uma irregularidade ex: 111211 -> 111111
i=1;j=2;k=3;
for l=2:length(ypred)-1
if ypred(i)~=ypred(j)
    if ypred(j)~=ypred(k)
        if ypred(i)==ypred(k)
            ypred(j)=ypred(k);
        end
    end
end
i=i+1;k=k+1;j=j+1;
ypred1=ypred;
end
erro1=cerror(ypred,tst.y);
%% para duas irregularidades ex: 11122111 -> 11111111
i=1;j=2;k=3;d=4;
for l=2:length(ypred)-2
if ypred(i)~=ypred(j)
    if ypred(j)~=ypred(k)
        if ypred(k)~=ypred(d)
            if ypred(i)==ypred(d)
                ypred(j)=ypred(i);
                ypred(k)=ypred(d);
            end
        end
    end
end
i=i+1;k=k+1;j=j+1;d=d+1;
end
i=1;j=2;k=3;d=4;
for l=2:length(ypred)-2
if ypred(i)~=ypred(j)
    if ypred(j)~=ypred(k)
        if ypred(k)~=ypred(d)
            if ypred(i)==ypred(d)
                ypred(j)=ypred(i);
                ypred(k)=ypred(d);
            end
        end
    end
end
```

```

        end
    end
    i=i+1;k=k+1;j=j+1;d=d+1;
    ypred2=ypred;
    end
    erro2=cerror(ypred,tst.y);
    %% para 3 irregularidades ex: 11113331111 -> 11111111111
    i=1;j=2;k=3;d=4;f=5;
    for l=3:length(ypred)-3
        if ypred(i)~=ypred(j)
            if ypred(j)==ypred(k)
                if ypred(k)==ypred(d)
                    if ypred(d)~=ypred(f)
                        if ypred(i)==ypred(f)
                            ypred(j)=ypred(i);
                            ypred(d)=ypred(f);
                            ypred(k)=ypred(j);
                        end
                    end
                end
            end
        end
    end
    i=i+1;k=k+1;j=j+1;d=d+1;f=f+1;
    end
    i=1;j=2;k=3;d=4;f=5;
    for l=3:length(ypred)-3
        if ypred(i)~=ypred(j)
            if ypred(j)~=ypred(k)
                if ypred(k)==ypred(d)
                    if ypred(d)~=ypred(f)
                        if ypred(i)==ypred(f)
                            ypred(j)=ypred(i);
                            ypred(d)=ypred(f);
                            ypred(k)=ypred(j);
                        end
                    end
                end
            end
        end
    end
    i=i+1;k=k+1;j=j+1;d=d+1;f=f+1;
    end
    i=1;j=2;k=3;d=4;f=5;
    for l=3:length(ypred)-3
        if ypred(i)~=ypred(j)
            if ypred(j)==ypred(k)
                if ypred(k)~=ypred(d)
                    if ypred(d)~=ypred(f)
                        if ypred(i)==ypred(f)
                            ypred(j)=ypred(i);
                            ypred(d)=ypred(f);
                            ypred(k)=ypred(j);
                        end
                    end
                end
            end
        end
    end
    i=i+1;k=k+1;j=j+1;d=d+1;f=f+1;
    end
    i=1;j=2;k=3;d=4;f=5;

```



```

for l=3:length(ypred)-3
if ypred(i)~=ypred(j)
    if ypred(j)~=ypred(k)
        if ypred(k)~=ypred(d)
            if ypred(d)~=ypred(f)
                if ypred(i)==ypred(f)
                    ypred(j)=ypred(i);
                    ypred(d)=ypred(f);
                    ypred(k)=ypred(j);
                end
            end
        end
    end
end
end
i=i+1;k=k+1;j=j+1;d=d+1;f=f+1;
ypred3=ypred;
end
erro3=cerror(ypred,tst.y);
%% para 4 irregularidades ex: 111444411111 -> 111111111111
i=1;j=2;k=3;d=4;f=5;g=6;
for l=3:length(ypred)-4
if ypred(i)~=ypred(j)
    if ypred(f)~=ypred(g)
        if ypred(i)==ypred(g)
            if ypred(j)~=ypred(k) || ypred(k)~=ypred(d) ||
ypred(d)~=ypred(f)
                ypred(j)=ypred(i);
                ypred(f)=ypred(g);
                ypred(k)=ypred(j);
                ypred(d)=ypred(f);
            elseif ypred(j)~=ypred(k) || ypred(k)~=ypred(d) ||
ypred(d)==ypred(f)
                ypred(j)=ypred(i);
                ypred(f)=ypred(g);
                ypred(k)=ypred(j);
                ypred(d)=ypred(f);
            elseif ypred(j)~=ypred(k) || ypred(k)==ypred(d) ||
ypred(d)~=ypred(f)
                ypred(j)=ypred(i);
                ypred(f)=ypred(g);
                ypred(k)=ypred(j);
                ypred(d)=ypred(f);
            elseif ypred(j)~=ypred(k) || ypred(k)==ypred(d) ||
ypred(d)==ypred(f)
                ypred(j)=ypred(i);
                ypred(f)=ypred(g);
                ypred(k)=ypred(j);
                ypred(d)=ypred(f);
            elseif ypred(j)==ypred(k) || ypred(k)~=ypred(d) ||
ypred(d)~=ypred(f)
                ypred(j)=ypred(i);
                ypred(f)=ypred(g);
                ypred(k)=ypred(j);
                ypred(d)=ypred(f);
            elseif ypred(j)==ypred(k) || ypred(k)~=ypred(d) ||
ypred(d)==ypred(f)
                ypred(j)=ypred(i);
                ypred(f)=ypred(g);
                ypred(k)=ypred(j);

```

```

        ypred(d)=ypred(f);
elseif ypred(j)==ypred(k) || ypred(k)==ypred(d) ||
ypred(d)~=ypred(f)
        ypred(j)=ypred(i);
        ypred(f)=ypred(g);
        ypred(k)=ypred(j);
        ypred(d)=ypred(f);
else ypred(j)==ypred(k) || ypred(k)==ypred(d) ||
ypred(d)==ypred(f)
        ypred(j)=ypred(i);
        ypred(f)=ypred(g);
        ypred(k)=ypred(j);
        ypred(d)=ypred(f);
end
end
end
end
i=i+1;k=k+1;j=j+1;d=d+1;f=f+1;g=g+1;
ypred4=ypred;
end
erro4=cerror(ypred,tst.y);
%% para 5 irregularidades ex: 11144444111111 -> 11111111111111
i=1;j=2;k=3;d=4;f=5;g=6;h=7;
for l=4:length(ypred)-5
if ypred(i)~=ypred(j)
    if ypred(g)~=ypred(h)
        if ypred(i)==ypred(h)
            if ypred(j)~=ypred(k) || ypred(k)~=ypred(d) ||
ypred(d)~=ypred(f) || ypred(f)~=ypred(g)
                ypred(j)=ypred(i);
                ypred(g)=ypred(h);
                ypred(k)=ypred(j);
                ypred(f)=ypred(g);
                ypred(d)=ypred(k);
            elseif ypred(j)~=ypred(k) || ypred(k)~=ypred(d) ||
ypred(d)~=ypred(f) || ypred(f)==ypred(g)
                ypred(j)=ypred(i);
                ypred(g)=ypred(h);
                ypred(k)=ypred(j);
                ypred(f)=ypred(g);
                ypred(d)=ypred(k);
            elseif ypred(j)~=ypred(k) || ypred(k)~=ypred(d) ||
ypred(d)==ypred(f) || ypred(f)~=ypred(g)
                ypred(j)=ypred(i);
                ypred(g)=ypred(h);
                ypred(k)=ypred(j);
                ypred(f)=ypred(g);
                ypred(d)=ypred(k);
            elseif ypred(j)~=ypred(k) || ypred(k)~=ypred(d) ||
ypred(d)==ypred(f) || ypred(f)==ypred(g)
                ypred(j)=ypred(i);
                ypred(g)=ypred(h);
                ypred(k)=ypred(j);
                ypred(f)=ypred(g);
                ypred(d)=ypred(k);
            elseif ypred(j)~=ypred(k) || ypred(k)==ypred(d) ||
ypred(d)~=ypred(f) || ypred(f)~=ypred(g)
                ypred(j)=ypred(i);
                ypred(g)=ypred(h);

```

```

        ypred(k)=ypred(j);
        ypred(f)=ypred(g);
        ypred(d)=ypred(k);
    elseif ypred(j)~=ypred(k) || ypred(k)==ypred(d) ||
ypred(d)~=ypred(f) || ypred(f)==ypred(g)
        ypred(j)=ypred(i);
        ypred(g)=ypred(h);
        ypred(k)=ypred(j);
        ypred(f)=ypred(g);
        ypred(d)=ypred(k);
    elseif ypred(j)~=ypred(k) || ypred(k)==ypred(d) ||
ypred(d)==ypred(f) || ypred(f)~=ypred(g)
        ypred(j)=ypred(i);
        ypred(g)=ypred(h);
        ypred(k)=ypred(j);
        ypred(f)=ypred(g);
        ypred(d)=ypred(k);
    elseif ypred(j)~=ypred(k) || ypred(k)==ypred(d) ||
ypred(d)==ypred(f) || ypred(f)==ypred(g)
        ypred(j)=ypred(i);
        ypred(g)=ypred(h);
        ypred(k)=ypred(j);
        ypred(f)=ypred(g);
        ypred(d)=ypred(k);
    elseif ypred(j)==ypred(k) || ypred(k)~=ypred(d) ||
ypred(d)~=ypred(f) || ypred(f)~=ypred(g)
        ypred(j)=ypred(i);
        ypred(g)=ypred(h);
        ypred(k)=ypred(j);
        ypred(f)=ypred(g);
        ypred(d)=ypred(k);
    elseif ypred(j)==ypred(k) || ypred(k)~=ypred(d) ||
ypred(d)~=ypred(f) || ypred(f)==ypred(g)
        ypred(j)=ypred(i);
        ypred(g)=ypred(h);
        ypred(k)=ypred(j);
        ypred(f)=ypred(g);
        ypred(d)=ypred(k);
    elseif ypred(j)==ypred(k) || ypred(k)~=ypred(d) ||
ypred(d)==ypred(f) || ypred(f)~=ypred(g)
        ypred(j)=ypred(i);
        ypred(g)=ypred(h);
        ypred(k)=ypred(j);
        ypred(f)=ypred(g);
        ypred(d)=ypred(k);
    elseif ypred(j)==ypred(k) || ypred(k)~=ypred(d) ||
ypred(d)==ypred(f) || ypred(f)==ypred(g)
        ypred(j)=ypred(i);
        ypred(g)=ypred(h);
        ypred(k)=ypred(j);
        ypred(f)=ypred(g);
        ypred(d)=ypred(k);
    elseif ypred(j)==ypred(k) || ypred(k)==ypred(d) ||
ypred(d)~=ypred(f) || ypred(f)~=ypred(g)
        ypred(j)=ypred(i);
        ypred(g)=ypred(h);
        ypred(k)=ypred(j);
        ypred(f)=ypred(g);
        ypred(d)=ypred(k);

```

```

        elseif ypred(j)==ypred(k) || ypred(k)==ypred(d) ||
ypred(d)~=ypred(f) || ypred(f)==ypred(g)
            ypred(j)=ypred(i);
            ypred(g)=ypred(h);
            ypred(k)=ypred(j);
            ypred(f)=ypred(g);
            ypred(d)=ypred(k);
        elseif ypred(j)==ypred(k) || ypred(k)==ypred(d) ||
ypred(d)==ypred(f) || ypred(f)~=ypred(g)
            ypred(j)=ypred(i);
            ypred(g)=ypred(h);
            ypred(k)=ypred(j);
            ypred(f)=ypred(g);
            ypred(d)=ypred(k);
        else ypred(j)==ypred(k) || ypred(k)==ypred(d) ||
ypred(d)==ypred(f) || ypred(f)==ypred(g)
            ypred(j)=ypred(i);
            ypred(g)=ypred(h);
            ypred(k)=ypred(j);
            ypred(f)=ypred(g);
            ypred(d)=ypred(k);
        end
    end
end
end
i=i+1;k=k+1;j=j+1;d=d+1;f=f+1;g=g+1;h=h+1;
ypred5=ypred;
end
erro5=cerror(ypred,tst.y);
end

```